Original-Programmierhandbuch BasicController

ecomat100
CR0411

Laufzeitsystem V03
CODESYS® V2.3

Deutsch

7391006_04_DE 2015-03-10

© ifm electronic gmbh
www.ifm.com

Inhaltsverzeichnis

1 Über diese Anleitung 5
1.1 Copyright ... 5
1.2 Übersicht: Dokumentations-Module für ecomatmobile-Geräte ... 6
1.3 CODESYS-Programmierhandbuch ... 6
1.4 Was bedeuten die Symbole und Formatierungen? ... 7
1.5 Wie ist diese Dokumentation aufgebaut? ... 8
1.6 Historie der Anleitung (CR041n) ... 9

2 Sicherheitshinweise 10
2.1 Beachten! ... 10
2.2 Welche Vorkenntnisse sind notwendig? ... 11
2.3 Anlaufverhalten der Steuerung ... 11

3 Systembeschreibung 12
3.1 Angaben zum Gerät ... 12
3.2 Hardware-Beschreibung ... 12
3.2.1 Hardware-Aufbau ... 13
3.2.2 Eingänge (Technologie) ... 14
3.2.3 Ausgänge (Technologie) ... 18
3.2.4 Hinweise zur Anschlussbelegung ... 24
3.2.5 Sicherheitshinweise zu Reed-Relais ... 24
3.2.6 Status-LED ... 25
3.3 Schnittstellen-Beschreibung ... 26
3.3.1 CAN-Schnittstellen ... 26
3.4 Software ... 27
3.4.1 Software-Module für das Gerät ... 27
3.4.2 Programmierhinweise für CODESYS-Projekte ... 30
3.4.3 Betriebszustände ... 34
3.4.4 Leistungsgrenzen des Geräts ... 37

4 Konfigurationen 39
4.1 Laufzeitsystem einrichten ... 39
4.1.1 Laufzeitsystem neu installieren ... 40
4.1.2 Laufzeitsystem aktualisieren ... 41
4.1.3 Installation verifizieren ... 41
4.2 Programmiersystem einrichten ... 42
4.2.1 Programmiersystem manuell einrichten ... 42
4.2.2 Programmiersystem über Templates einrichten ... 46
4.3 Funktionskonfiguration, allgemein ... 46
4.3.1 Systemvariablen ... 46
4.4 Funktionskonfiguration der Ein- und Ausgänge ... 47
4.4.1 Konfiguration der Ein- und Ausgänge (Voreinstellung) ... 47
4.4.2 Eingänge konfigurieren ... 48
4.4.3 Ausgänge konfigurieren ... 52
4.5 Variablen ... 56
4.5.1 Retain-Variablen ... 57
4.5.2 Netzwerkvariablen ... 58

© ifm electronic gmbh www.ifm.com

5 ifm-Funktionselemente 59

5.1 ifm-Bibliotheken für das Gerät CR0411 59
5.1.1 Bibliothek ifm_CR0411_V03yyzz.LIB 60
5.1.2 Bibliothek ifm_RAWCan_NT_Vxxyyzz.LIB 61
5.1.3 Bibliothek ifm_CANopen_NT_Vxxyyzz.LIB 62
5.1.4 Bibliothek ifm_J1939_NT_Vxxyyzz.LIB 63
5.1.5 Bibliothek ifm_CR0411_util_Vxxyyzz.LIB 64
5.2 ifm-Bausteine für das Gerät CR0411 65
5.2.1 Baustein-Ausgänge 66
5.2.2 Bausteine: RAW-CAN (Layer 2) 67
5.2.3 Bausteine: CANopen 93
5.2.4 Bausteine: SAE J1939 138
5.2.5 Bausteine: Eingangswerte verarbeiten 170
5.2.6 Bausteine: Ausgangsfunktionen 180
5.2.7 Bausteine: System 194

6 Diagnose und Fehlerbehandlung 212

6.1 Diagnose 212
6.2 Fehler 212
6.3 Reaktion auf Fehlermeldungen 213
6.3.1 Beispielablauf für Reaktion auf Fehlermeldungen 213
6.4 CAN / CANopen: Fehler und Fehlerbehandlung 213

7 Anhang 214

7.1 Systemmerker 214
7.2 Adressbelegung und E/A-Betriebsarten 214
7.2.1 Adressbelegung Ein-/Ausgänge 215
7.2.2 Mögliche Betriebsarten Ein-/Ausgänge 216
7.3 Fehler-Tabellen 218
7.3.1 Fehlermerker 218
7.3.2 Fehler: CAN / CANopen 218

8 Begriffe und Abkürzungen 220

9 Index 234

10 Notizen • Notes • Notes 238

11 ifm weltweit • ifm worldwide • ifm à l'échelle internationale 243

© ifm electronic gmbh www.ifm.com

1 Über diese Anleitung

Inhalt

Copyright 5
Übersicht: Dokumentations-Module für ecomatmobile-Geräte 6
CODESYS-Programmierhandbuch 6
Was bedeuten die Symbole und Formatierungen? 7
Wie ist diese Dokumentation aufgebaut? 8
Historie der Anleitung (CR041n) 9

202

1.1 Copyright

6088

© Alle Rechte bei **ifm electronic gmbh**. Vervielfältigung und Verwertung dieser Anleitung, auch auszugsweise, nur mit Zustimmung der **ifm electronic gmbh**.

Alle auf unseren Seiten verwendeten Produktnamen, -Bilder, Unternehmen oder sonstige Marken sind Eigentum der jeweiligen Rechteinhaber:

- AS-i ist Eigentum der AS-International Association, (→ www.as-interface.net)
- CAN ist Eigentum der CiA (CAN in Automation e.V.), Deutschland (→ www.can-cia.org)
- CODESYS™ ist Eigentum der 3S – Smart Software Solutions GmbH, Deutschland (→ www.codesys.com)
- DeviceNet™ ist Eigentum der ODVA™ (Open DeviceNet Vendor Association), USA (→ www.odva.org)
- EtherNet/IP® ist Eigentum der →ODVA™
- IO-Link® (→ www.io-link.com) ist Eigentum der →PROFIBUS Nutzerorganisation e.V., Deutschland
- Microsoft® ist Eigentum der Microsoft Corporation, USA (→ www.microsoft.com)
- PROFIBUS® ist Eigentum der PROFIBUS Nutzerorganisation e.V., Deutschland (→ www.profibus.com)
- PROFINET® ist Eigentum der →PROFIBUS Nutzerorganisation e.V., Deutschland
- Windows® ist Eigentum der →Microsoft Corporation, USA

© ifm electronic gmbh

1.2 Übersicht: Dokumentations-Module für ecomatmobile-Geräte

17405

Die Dokumentation für ecomatmobile-Geräte besteht aus folgenden Modulen:

1.	Datenblatt
Inhalt:	Technische Daten in Tabellenform
Quelle:	→ www.ifm.com > Land wählen > [Datenblattsuche] > CR0411 > [Technische Daten im PDF-Format]

2.	Montageanleitung / Betriebsanleitung
Inhalt:	Anleitung für Montage, elektrische Installation, (Inbetriebnahme*), Technische Daten
Quelle:	Anleitung wird mit dem Gerät mitgeliefert Auch zu finden auf der ifm-Homepage: → www.ifm.com > Land wählen > [Datenblattsuche] > CR0411 > [Betriebsanleitungen]

3.	Programmierhandbuch + Online-Hilfe
Inhalt:	Beschreibung der Konfiguration und der Funktionen der Geräte-Software
Quelle:	→ www.ifm.com > Land wählen > [Datenblattsuche] > CR0411 > [Betriebsanleitungen]

4.	Systemhandbuch "Know-How ecomatmobile"
Inhalt:	Hintergrundwissen zu folgenden Themen: • Übersicht Templates und Demo-Programme • CAN, CANopen • Ausgänge steuern • User-Flash-Speicher • Visualisierungen • Übersicht Dateien und Bibliotheken
Quelle:	→ www.ifm.com > Land wählen > [Datenblattsuche] > CR0411 > [Betriebsanleitungen]

*) Die in Klammern gesetzten Beschreibungen sind nur in den Anleitungen bestimmter Geräte enthalten.

1.3 CODESYS-Programmierhandbuch

17542

Im ergänzenden "Programmierhandbuch CODESYS V2.3" der 3S GmbH erhalten Sie weitergehende Informationen über die Nutzung des Programmiersystems.
Dieses Handbuch steht auf der ifm-Homepage als kostenloser Download zur Verfügung:
→ www.ifm.com > Land wählen > [Service] > [Download] > [Systeme für mobile Arbeitsmaschinen]
Handbücher und Online-Hilfen für ecomatmobile finden Sie auch hier:
→ ecomatmobile-DVD "Software, tools and documentation"

© ifm electronic gmbh www.ifm.com

1.4 Was bedeuten die Symbole und Formatierungen?

203

Folgende Symbole oder Piktogramme verdeutlichen Ihnen unsere Hinweise in unseren Anleitungen:

⚠ WARNUNG

Tod oder schwere irreversible Verletzungen sind möglich.

⚠ VORSICHT

Leichte reversible Verletzungen sind möglich.

ACHTUNG

Sachschaden ist zu erwarten oder möglich.

	Wichtige Hinweise auf Fehlfunktionen oder Störungen
	Weitere Hinweise
► ...	Handlungsaufforderung
> ...	Reaktion, Ergebnis
→ ...	"siehe"
abc	Querverweis
123 0x123 0b010	Dezimalzahl Hexadezimalzahl Binärzahl
[...]	Bezeichnung von Tasten, Schaltflächen oder Anzeigen

© ifm electronic gmbh

1.5 Wie ist diese Dokumentation aufgebaut?

204
1508

Diese Dokumentation ist eine Kombination aus verschiedenen Anleitungstypen. Sie ist eine Lernanleitung für den Einsteiger, aber gleichzeitig auch eine Nachschlageanleitung für den versierten Anwender. Dieses Dokument richtet sich an die Programmierer der Anwendungen.

Und so finden Sie sich zurecht:

- Um gezielt zu einem bestimmten Thema zu gelangen, benutzen Sie bitte das Inhaltsverzeichnis.
- Mit dem Stichwortregister "Index" gelangen Sie ebenfalls schnell zu einem gesuchten Begriff.
- Am Anfang eines Kapitels geben wir Ihnen eine kurze Übersicht über dessen Inhalt.
- Abkürzungen und Fachbegriffe → Anhang.

Bei Fehlfunktionen oder Unklarheiten setzen Sie sich bitte mit dem Hersteller in Verbindung:
→ www.ifm.com > Land wählen > [Kontakt].

Wir wollen immer besser werden! Jeder eigenständige Abschnitt enthält in der rechten oberen Ecke eine Identifikationsnummer. Wenn Sie uns über Unstimmigkeiten unterrichten wollen, dann nennen Sie uns bitte diese Nummer zusammen mit Titel und Sprache dieser Dokumentation. Vielen Dank für Ihre Unterstützung!

Im Übrigen behalten wir uns Änderungen vor, so dass sich Abweichungen vom Inhalt der vorliegenden Dokumentation ergeben können. Die aktuelle Version finden Sie auf der **ifm**-Homepage:
→ www.ifm.com > Land wählen > [Datenblattsuche] > (Artikel-Nr.) > [Betriebsanleitungen]

© ifm electronic gmbh www.ifm.com

1.6 Historie der Anleitung (CR041n)

15905

Was hat sich wann in dieser Anleitung geändert? Ein Überblick:

Datum	Thema	Änderung
2014-04-28	diverse FBs	Beschreibung FB-Eingang CHANNEL präzisiert
2014-04-29	FB CAN_REMOTE_RESPONSE	Beschreibung FB-Eingang ENABLE präzisiert
2014-05-12	Leistungsgrenzen CAN	Leistungsgrenzen ergänzt für CAN, CANopen und CAN J1939
2014-06-30	Name der Dokumentation	"Systemhandbuch" umbenannt zu "Programmierhandbuch"
2014-08-08	FB PERIOD	ergänzt um Betriebsart "Phasenmessung" (ab LZS V03.02.zz)
2014-08-08	FB H_BRIDGE	Warnung vor Notbremsbetrieb
2014-08-26	Beschreibung Eingänge, Ausgänge	highside / lowside ersetzt durch plusschaltend / minusschaltend
2014-10-15	FB H_BRIDGE	Grafik Prinzipschaltung korrigiert
2014-11-12	Kapitel "Ausgänge (Technologie)"	Abschnitt "Diagnose der binären Ausgänge" ergänzt oder korrigiert
2015-01-13	Dokumentationsstruktur Fehlercodes, Systemmerker	• Fehlermerker: nur noch im Anhang, Kapitel ***Systemmerker*** • CAN / CANopen Fehler und Fehlerbehandlung: nur noch im Systemhandbuch "Know-How" • Fehlercodes, EMCY-Codes: nun im Anhang, Kapitel ***Fehler-Tabellen***
2015-03-10	Verfügbarer Speicher	Darstellung verbessert

© ifm electronic gmbh

2 Sicherheitshinweise

Inhalt

Beachten! 10
Welche Vorkenntnisse sind notwendig? 11
Anlaufverhalten der Steuerung 11

213

2.1 Beachten!

6091
11212

Mit den in dieser Anleitung gegebenen Informationen, Hinweisen und Beispielen werden keine Eigenschaften zugesichert. Die abgebildeten Zeichnungen, Darstellungen und Beispiele enthalten weder Systemverantwortung noch anwendungsspezifische Besonderheiten.

- ► Die Sicherheit der Maschine/Anlage muss auf jeden Fall eigenverantwortlich durch den Hersteller der Maschine/Anlage gewährleistet werden.
- ► Beachten Sie die nationalen Vorschriften des Landes, in welchem die Maschine/Anlage in Verkehr gebracht werden soll!

⚠ WARNUNG

Bei Nichtbeachten der Hinweise in dieser Anleitung sind Sach- oder Körperschäden möglich!
Die **ifm electronic gmbh** übernimmt hierfür keine Haftung.

- ► Die handelnde Person muss vor allen Arbeiten an und mit diesem Gerät die Sicherheitshinweise und die betreffenden Kapitel dieser Anleitung gelesen und verstanden haben.
- ► Die handelnde Person muss zu Arbeiten an der Maschine/Anlage autorisiert sein.
- ► Die handelnde Person muss für die auszuführende Arbeit über die erforderliche Ausbildung und Qualifikation verfügen.
- ► Beachten Sie die Technischen Daten der betroffenen Geräte!
  Das aktuelle Datenblatt finden Sie auf der **ifm**-Homepage:
  → www.ifm.com > Land wählen > [Datenblattsuche] > (Artikel-Nr.) > [Technische Daten im PDF-Format]
- ► Beachten Sie die Montage- und Anschlussbedingungen sowie die bestimmungsgemäße Verwendung der betroffenen Geräte!
  → mitgelieferte Montageanleitung oder auf der **ifm**-Homepage:
  → www.ifm.com > Land wählen > [Datenblattsuche] > (Artikel-Nr.) > [Betriebsanleitungen]
- ► Beachten Sie die Korrekturen und Hinweise in den "Release-Notes" zur vorhandenen Hardware, Software und Dokumentation auf der **ifm**-Homepage:
  → www.ifm.com > Land wählen > [Datenblattsuche] > (Artikel-Nr.) > [Betriebsanleitungen]

© ifm electronic gmbh

2.2 Welche Vorkenntnisse sind notwendig?

215

Das Dokument richtet sich an Personen, die über Kenntnisse der Steuerungstechnik und SPS-Programmierkenntnisse mit IEC 61131-3 verfügen.

Zum Programmieren der SPS sollten die Personen zusätzlich mit der Software CODESYS vertraut sein.

Das Dokument richtet sich an Fachkräfte. Dabei handelt es sich um Personen, die aufgrund ihrer einschlägigen Ausbildung und ihrer Erfahrung befähigt sind, Risiken zu erkennen und mögliche Gefährdungen zu vermeiden, die der Betrieb oder die Instandhaltung eines Produkts verursachen kann. Das Dokument enthält Angaben zum korrekten Umgang mit dem Produkt.

Lesen Sie dieses Dokument vor dem Einsatz, damit Sie mit Einsatzbedingungen, Installation und Betrieb vertraut werden. Bewahren Sie das Dokument während der gesamten Einsatzdauer des Gerätes auf.

Befolgen Sie die Sicherheitshinweise.

2.3 Anlaufverhalten der Steuerung

15233

⚠ WARNUNG

Gefahr durch unbeabsichtigtes und gefährliches Anlaufen von Maschinen- oder Anlagenteilen!

- ► Der Programmierer muss bei der Programmerstellung verhindern, dass nach Auftreten eines Fehlers (z.B. NOT-HALT) und der anschließenden Fehlerbeseitigung unbeabsichtigt Maschinen- oder Anlagenteile gefährlich anlaufen können!
  ⇨ Wiederanlaufsperre realisieren!
- ► Dazu im Fehlerfall die in Frage kommenden Ausgänge im Programm logisch abschalten!

Ein Wiederanlauf kann z.B. verursacht werden durch:
- Spannungswiederkehr nach Spannungsausfall
- Reset nach Watchdog-Ansprechen wegen zu langer Zykluszeit
- Fehlerbeseitigung nach NOT-HALT

So erreichen Sie sicheres Verhalten der Steuerung:

- ► Spannungsversorgung im Anwendungsprogramm überwachen.
- ► Im Fehlerfall alle relevanten Ausgänge im Anwendungsprogramm ausschalten.
- ► Aktuatoren, die zu gefahrbringenden Bewegungen führen können, zusätzlich im Anwendungsprogramm überwachen (Feedback).

6827

- ► Relaiskontakte, die zu gefahrbringenden Bewegungen führen können, zusätzlich im Anwendungsprogramm überwachen (Feedback).
- ► Bei Bedarf im Anwendungsprojekt sicherstellen, dass verschweißte Relaiskontakte keine gefahrbringenden Bewegungen auslösen oder fortführen können.

© ifm electronic gmbh www.ifm.com

3 Systembeschreibung

Inhalt

Angaben zum Gerät 12
Hardware-Beschreibung 12
Schnittstellen-Beschreibung 26
Software 27

975

3.1 Angaben zum Gerät

14551

Diese Anleitung beschreibt aus der Gerätefamilie für den mobilen Einsatz, ecomat*mobile* der ifm electronic gmbh:

- BasicController: CR0411

3.2 Hardware-Beschreibung

Inhalt

Hardware-Aufbau 13
Eingänge (Technologie) 14
Ausgänge (Technologie) 18
Hinweise zur Anschlussbelegung 24
Sicherheitshinweise zu Reed-Relais 24
Status-LED 25

14081

© ifm electronic gmbh

3.2.1 Hardware-Aufbau

14561

Grundgerät ohne Abdeckung
Schutzart IP 20

Verfügbarer Speicher

13736

FLASH-Speicher

13053

FLASH-Speicher (nichtflüchtiger, langsamer Speicher) insgesamt im Gerät vorhanden	1 536 kByte

Davon sind folgende Speicherbereiche reserviert für ...

maximale Größe für das Anwendungsprogramm	512 kByte
Daten außerhalb des Anwendungsprogramms Daten mit FB FLASH_READ lesen (bei Files: abzüglich 128 Byte für Header)	64 kByte

Der verbleibende Speicher ist reserviert für system-interne Zwecke.

SRAM

14027

SRAM (flüchtiger, schneller Speicher) insgesamt im Gerät vorhanden SRAM steht hier allgemein für alle Arten von flüchtigen, schnellen Speichern.	592 kByte

Davon sind folgende Speicherbereiche reserviert für ...

vom Anwendungsprogramm reservierte Daten	128 kByte

Der verbleibende Speicher ist reserviert für system-interne Zwecke.

FRAM

2262

FRAM (nichtflüchtiger, schneller Speicher) insgesamt im Gerät vorhanden FRAM steht hier allgemein für alle Arten von nichtflüchtigen, schnellen Speichern.	2 kByte

Davon sind folgende Speicherbereiche reserviert für ...

im Anwendungsprogramm als VAR_RETAIN deklarierte Variablen	128 Byte
fest als remanent definierte Merker (%MB0…127)	128 Byte

Der verbleibende Speicher ist reserviert für system-interne Zwecke.

© ifm electronic gmbh www.ifm.com

3.2.2 Eingänge (Technologie)

Inhalt

Analog-Eingänge 14
Binär-Eingänge 15
Eingangsgruppe I0 (IN0...IN3) 16
Eingangsgruppe I1 (IN4...IN7) 16

14090

Analog-Eingänge

15444

Die Analog-Eingänge können über das Anwendungsprogramm konfiguriert werden. Der Messbereich kann zwischen folgenden Bereichen umgeschaltet werden:

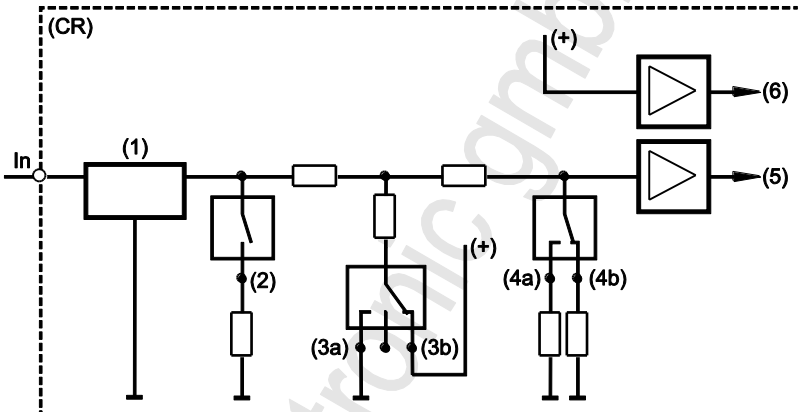
- Stromeingang 0...20 mA
- Spannungseingang 0...10 V
- Spannungseingang 0...32 V
- Widerstandsmessung 16...30 000 Ω (Messung gegen GND)

Die Spannungsmessung kann auch ratiometrisch erfolgen (0...1000 ‰, über FBs einstellbar). Das bedeutet, ohne zusätzliche Referenzspannung können Potentiometer oder Joysticks ausgewertet werden. Ein Schwanken der Versorgungsspannung hat auf diesen Messwert keinen Einfluss.

Alternativ kann ein Analog-Kanal auch binär ausgewertet werden.

Bei ratiometrischer Messung müssen die angeschlossenen Sensoren mit VBB_S des Geräts versorgt werden. Dadurch werden Fehlmessungen durch Spannungsverschiebungen vermieden.

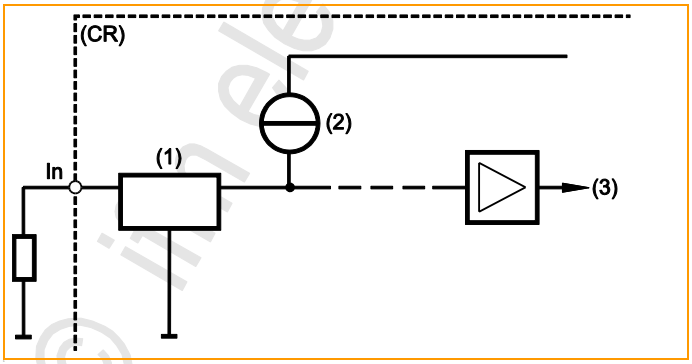
8971



In = Anschluss Multifunktions-Eingang n
(CR) = Gerät
(1) = Eingangsfilter
(2) = analoge Strommessung
(3a) = Binär-Eingang plus-schaltend
(3b) = Binär-Eingang minus-schaltend
(4a) = analoge Spannungsmessung 0...10 V
(4b) = analoge Spannungsmessung 0...32 V
(5) = Spannung
(6) = Referenz-Spannung

Grafik: Prinzipschaltung Multifunktions-Eingang

8972



Grafik: Prinzipschaltung Widerstandsmess-Eingang

In = Anschluss Widerstandsmess-Eingang n
(CR) = Gerät
(1) = Eingangsfilter
(2) = Konstantstromquelle
(3) = Spannung

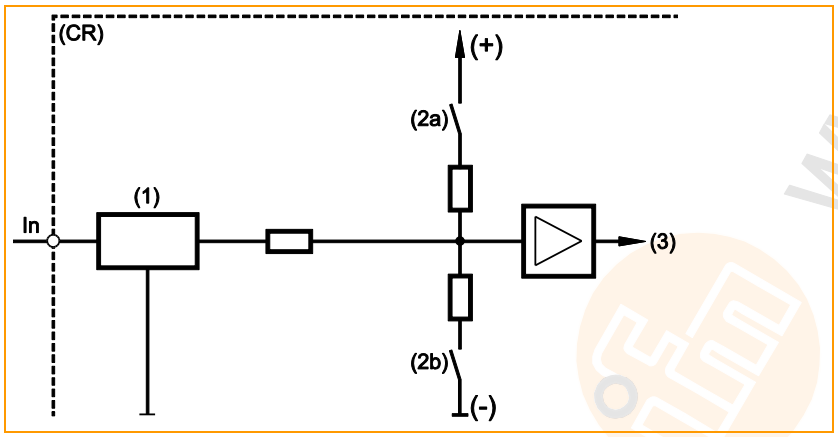
© ifm electronic gmbh www.ifm.com

Binär-Eingänge

1015
7345

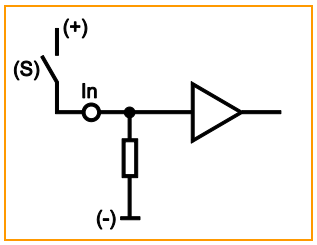
Der Binär-Eingang kann in folgenden Modi betrieben werden:
- binärer Eingang plus-schaltend (BL) für positives Gebersignal
- binärer Eingang minus-schaltend (BH) für negatives Gebersignal

Je nach Gerät können auch die Binär-Eingänge unterschiedlich konfiguriert werden. Neben den Schutzmechanismen gegen Störungen werden die Binär-Eingänge intern über eine Analogstufe ausgewertet. Das ermöglicht die Diagnose der Eingangssignale. Im Anwendungsprogramm steht das Schaltsignal aber direkt als Bit-Information zur Verfügung.



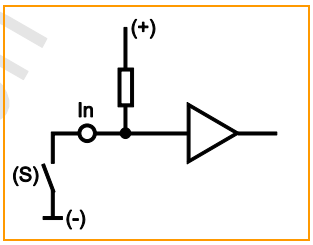
In = Anschluss Binär-Eingang n
(CR) = Gerät
(1) = Input-Filter
(2a) = Eingang minus-schaltend
(2b) = Eingang plus-schaltend
(3) = Spannung

Grafik: Prinzipschaltung Binär-Eingang minus-schaltend / plus-schaltend für negative und positive Gebersignale



In = Anschluss Binär-Eingang n
(S) = Sensor

Prinzipschaltung Binär-Eingang plus-schaltend (BL)
für positives Sensorsignal:
Eingang = offen ⇨ Signal = Low (Supply)



In = Anschluss Binär-Eingang n
(S) = Sensor

Prinzipschaltung Binär-Eingang minus-schaltend (BH)
für negatives Sensorsignal:
Eingang = offen ⇨ Signal = High (GND)

Bei einem Teil dieser Eingänge (→ Datenblatt) kann das Potential gewählt werden, gegen das geschaltet wird.

© ifm electronic gmbh

Eingangsgruppe I0 (IN0...IN3)

14568

Bei diesen Eingängen handelt es sich um eine Gruppe von Multifunktionskanälen.

Jeder einzelne dieser Eingänge ist wahlweise wie folgt konfigurierbar:
- analoger Eingang 0...20 mA
- analoger Eingang 0...10 V
- analoger Eingang 0...32 V
- Spannungsmessung ratiometrisch 0...1000 ‰
- binärer Eingang plus-schaltend (BL) für positives Gebersignal (mit/ohne Diagnose)
- binärer Eingang, minus-schaltend (BH) für negatives Gebersignal
- schneller Eingang für z.B. Inkrementalgeber und Frequenz- oder Periodendauermessung

→ Kapitel ***Mögliche Betriebsarten Ein-/Ausgänge*** (→ Seite 216)

Diagnosefähige Sensoren nach NAMUR können ausgewertet werden.

Alle Eingänge zeigen das gleiche Verhalten bei Funktion und Diagnose.

ⓘ Detaillierte Beschreibung → Kapitel ***Adressbelegung Ein-/Ausgänge*** (→ Seite 215)

Die Konfiguration jedes einzelnen Eingangs erfolgt über das Anwendungsprogramm:
- FB ***INPUT*** (→ Seite 176) > Eingang MODE
- FBs ***FASTCOUNT*** (→ Seite 171), ***INC_ENCODER*** (→ Seite 173) oder ***PERIOD*** (→ Seite 178)

> Werden die Analogeingänge auf Strommessung konfiguriert, wird bei Überschreiten des Endwertes (23 mA für ≥ 40 ms) in den sicheren Spannungsmessbereich (0...32 V DC) geschaltet und im FB INPUT der Ausgang RESULT entsprechend gesetzt. Nach etwa einer Sekunde schaltet der Eingang selbsttätig auf den Strommessbereich zurück.

Eingangsgruppe I1 (IN4...IN7)

14569

Bei diesen Eingängen handelt es sich um eine Gruppe von Multifunktionskanälen.

Jeder einzelne dieser Eingänge ist wahlweise wie folgt konfigurierbar:
- binärer Eingang plus-schaltend (BL) für positives Gebersignal
- Eingang für Widerstandsmessung (z.B. Temperatursensoren oder Tankgeber)

→ Kapitel ***Mögliche Betriebsarten Ein-/Ausgänge*** (→ Seite 216)

Diagnosefähige Sensoren nach NAMUR können ausgewertet werden.

► Die Konfiguration jedes einzelnen Eingangs erfolgt über das Anwendungsprogramm:
  - FB ***INPUT*** (→ Seite 176) > Eingang MODE

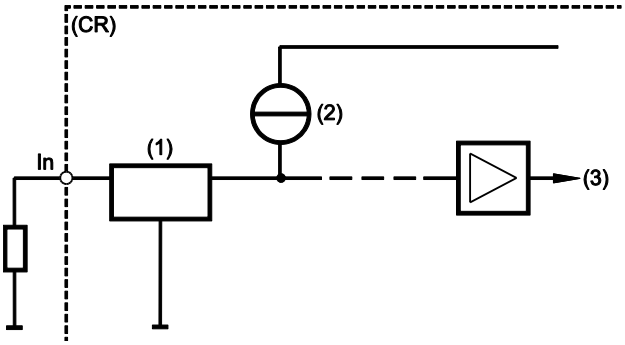
© ifm electronic gmbh www.ifm.com

Widerstandsmessung

9773

Typische Sensoren an diesen Eingängen:
- Tankpegel
- Temperatur (PT1000, NTC)

8972

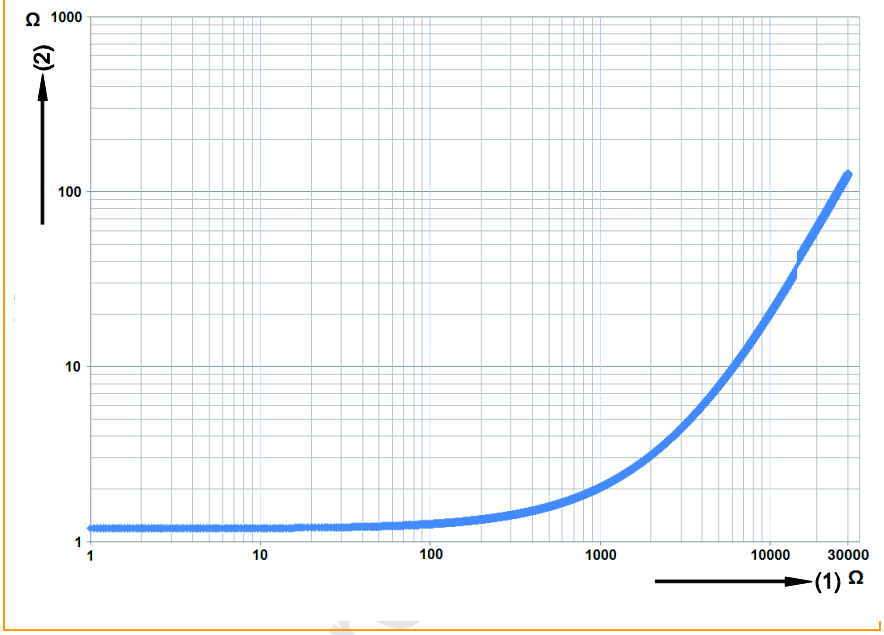


Grafik: Prinzipschaltung Widerstandsmess-Eingang

In = Anschluss Widerstandsmess-Eingang n
(CR) = Gerät
(1) = Eingangsfilter
(2) = Konstantstromquelle
(3) = Spannung

8970

Bei diesem Gerät ist die Auflösung nicht linear abhängig vom Widerstandswert, → Grafik:



Grafik: Auflösung abhängig vom Widerstandswert

(1) = Widerstandswert am Eingang
(2) = Auflösung

Um wieviel Ohm ändert sich der Messwert, wenn sich das Signal des A/D-Wandlers am Eingang um 1 ändert? Beispiele:
- Im Bereich 1...100 Ω beträgt die Auflösung 1,2 Ω.
- Im Bereich bei 1 kΩ beträgt die Auflösung ca. 2 Ω.
- Im Bereich bei 2 kΩ beträgt die Auflösung ca. 3 Ω.
- Im Bereich bei 3 kΩ beträgt die Auflösung ca. 6 Ω.
- Im Bereich bei 6 kΩ beträgt die Auflösung ca. 10 Ω.
- Im Bereich bei 10 kΩ beträgt die Auflösung ca. 11 Ω
- Im Bereich bei 20 kΩ beträgt die Auflösung ca. 60 Ω.

© ifm electronic

3.2.3 Ausgänge (Technologie)

Inhalt

Schutzfunktionen der Ausgänge 18
Ausgangsgruppe Q0 (OUT0...OUT3) 20
Ausgangsgruppe Q1 (OUT4...OUT7) 22

14093

Schutzfunktionen der Ausgänge

15248

Die Ausgänge dieses Geräts sind in Grenzen gegen Überlast und Kurzschluss geschützt.
→ Datenblatt

Definition: Überlast

15249

Überlast kann nur an einem Ausgang mit Strommessung erkannt werden.

Überlast ist definiert als ...
"nominaler Maximalstrom laut Datenblatt + 12,5 %".

Definition: Kurzschluss

15644

Ein Kurzschluss kann an allen diagnosefähigen Ausgängen erkannt werden und ist wie folgt definiert:

Kurzschluss ist definiert als ...
"Absinken der Ausgangsspannung unter 93,5 % (± 2,0 %) der zugehörigen Versorgungsspannung."

> Ein Schluss gegen Masse kann nur erkannt werden bei Ausgang = TRUE.

© ifm electronic gmbh

Reaktion der Ausgänge auf Überlast oder Kurzschluss

15251

Eigenschutz des Ausgangs

15253

Unabhängig von der Betriebsart des Ausgangs und der Fehlererkennung schützt sich die Hardware selbst. Bei zu hoher thermischer Belastung (durch Kurzschluss oder Überlast) beginnt der Ausgangstreiber zu takten.

Bei zu lange andauerndem Takten des Ausgangs (mehrere Stunden) kann der Treiber beschädigt werden!

Wir empfehlen deshalb:
Diagnosefähige Ausgänge des Geräts unbedingt mit folgenden Einstellungen betreiben, da hier die Software zusätzlich die Treiber durch Abschalten schützt:
• FB ***OUTPUT*** (→ Seite 183) > Eingang MODE = 16
Dies ist auch dann voreingestellt, wenn nur die Merker in der Steuerungskonfiguration verwendet werden.

Reaktion abhängig von Betriebsart des Ausgangs

15252

Im Falle von Überlast oder Kurzschluss hängt das Verhalten des Ausgangs von dessen Betriebsart ab (→ FB ***OUTPUT*** (→ Seite 183) > Eingang MODE):

- MODE=2: binary output plus-schaltend: keine Diagnose und kein Schutz
  > der Ausgang wird weiter betrieben.
- MODE=15: binary output plus-schaltend with diagnosis
  > Fehler wird erkannt und vom FB OUTPUT am Ausgang RESULT gemeldet:
  z.B.: RESULT = 128, 141, 142 oder 145.
  Das hängt vom Ausgangstyp und dem Strom oder der Spannung am Ausgang ab.
  Der Programmierer kann im Programm auf den Fehler reagieren.
- MODE=16: binary output plus-schaltend with diagnosis and protection
  > Fehler wird erkannt und vom FB OUTPUT am Ausgang RESULT gemeldet.
  > Der betreffende Ausgang wird abgeschaltet.
  > Der logische Zustand des Ausgangs bleibt davon unverändert!

Reaktion bei Einsatz von PWM oder CURRENT_CONTROL

15254

Anders verhält es sich bei Einsatz der FBs PWM oder CURRENT_CONTROL:
Hier gibt es keine Diagnose. Der →Eigenschutz des Ausgangs wird aktiv.

► Bei Ausgängen mit Stromrücklesung:
Im Anwendungsprogramm den typischen Strom für den Ausgang abfragen!
Hier ist der Programmierer verantwortlich, auf das Ereignis zu reagieren.

© ifm electronic gmbh www.ifm.com

Ausgangsgruppe Q0 (OUT0...OUT3)

15908

Bei diesen Ausgängen handelt es sich um eine Gruppe von Multifunktionskanälen.

Jeder einzelne dieser Ausgänge ist wahlweise wie folgt konfigurierbar:
- binärer Ausgang, plus-schaltend (BH) mit/ohne Diagnosefunktion
- analoger Ausgang, stromgeregelt (PWMi)
- analoger Ausgang mit Pulsweitenmodulation (PWM)

→ Kapitel ***Mögliche Betriebsarten Ein-/Ausgänge*** (→ Seite 216)

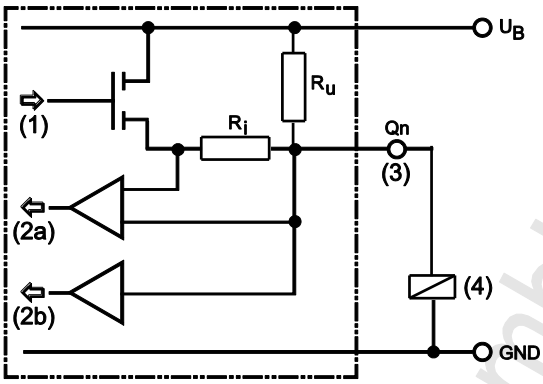
► Die Konfiguration jedes einzelnen Ausgangs erfolgt über das Anwendungsprogramm:
→ FB ***OUTPUT*** (→ Seite 183) > Eingang MODE
PWM-Ausgang: → FB ***PWM1000*** (→ Seite 185)
Stromregelung und Anzeigen der Lastströme → FB ***CURRENT_CONTROL*** (→ Seite 181)

► Zu den Grenzwerten unbedingt das Datenblatt beachten!

Diagnose: binäre Ausgänge (via Strom- und Spannungsmessung)

19433
19434

Die Diagnose dieser Ausgänge erfolgt über eine interne Strom- und Spannungsmessung im Ausgang:



Grafik: Prinzipschaltung
(1) Ausgangskanal
(2a) Rücklesekanal für Diagnose via Strommessung
(2b) Rücklesekanal für Diagnose via Spannungsmessung
(3) Anschluss Ausgang
(4) Last

Diagnose: Überlast

19437
15249

Überlast kann nur an einem Ausgang mit Strommessung erkannt werden.

Überlast ist definiert als ...
"nominaler Maximalstrom laut Datenblatt + 12,5 %".

Diagnose: Leiterbruch (via Strommessung)

19435
19400

Eine Leiterbruch-Erkennung erfolgt über den Rücklesekanal. Bei geschaltetem Ausgang (Qn=TRUE) wird dann ein Leiterbruch erkannt, wenn über den Widerstand R_i kein Strom fließt (keine Spannung abfällt). Ohne den Leiterbruch fließt durch den Längswiderstand R_i der Laststrom und erzeugt damit einen Spannungsabfall, der über den Rücklesekanal ausgewertet wird.

© ifm electronic gmbh www.ifm.com

Diagnose: Leiterbruch (via Spannungsmessung)

19436
19404

Eine Leiterbruch-Erkennung erfolgt über den Rücklesekanal. Bei gesperrtem Ausgang (Qn=FALSE) wird dann ein Leiterbruch erkannt, wenn der Widerstand R_u den Rücklesekanal auf HIGH-Potential (VBB) zieht. Ohne den Leiterbruch würde die niederohmige Last (R_L < 10 kOhm) LOW (logisch 0) erzwingen.

Diagnose: Kurzschluss

19405

Eine Kurzschluss-Erkennung erfolgt über den Rücklesekanal. Bei geschaltetem Ausgang (Qn=TRUE) wird dann ein Kurzschluss gegen GND erkannt, wenn der Rücklesekanal auf LOW-Potential (GND) gezogen wird.

© ifm electronic gmbh www.ifm.com

Ausgangsgruppe Q1 (OUT4...OUT7)

15822

Bei diesen Ausgängen handelt es sich um eine Gruppe von Multifunktionskanälen.

Jeder einzelne dieser Ausgänge ist wahlweise wie folgt konfigurierbar:
- binärer Ausgang, plus-schaltend (BH), teilweise auch minus-schaltend (BL)
- analoger Ausgang, stromgeregelt (PWMi)
- analoger Ausgang mit Pulsweitenmodulation (PWM), teilweise als H-Brücke

→ Kapitel ***Mögliche Betriebsarten Ein-/Ausgänge*** (→ Seite 216)

► Die Konfiguration jedes einzelnen Ausgangs erfolgt über das Anwendungsprogramm:
→ FB ***OUTPUT*** (→ Seite 183) > Eingang MODE
Stromregelung und Anzeigen der Lastströme → FB ***CURRENT_CONTROL*** (→ Seite 181)
PWM-Ausgang: → FB ***PWM1000*** (→ Seite 185)
H-Brücke steuern → FB ***H_BRIDGE*** (→ Seite 189)

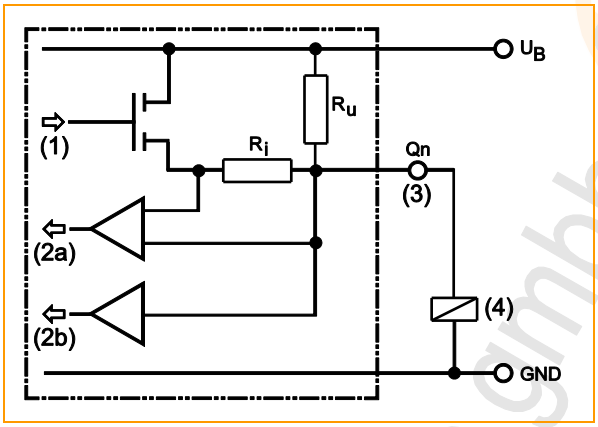
Bei Einsatz der H-Brücke wird die Stromregelung nicht unterstützt.

► Zu den Grenzwerten unbedingt das Datenblatt beachten!

Diagnose: binäre Ausgänge (via Strom- und Spannungsmessung)

19433
19434

Die Diagnose dieser Ausgänge erfolgt über eine interne Strom- und Spannungsmessung im Ausgang:



Grafik: Prinzipschaltung
(1) Ausgangskanal
(2a) Rücklesekanal für Diagnose via Strommessung
(2b) Rücklesekanal für Diagnose via Spannungsmessung
(3) Anschluss Ausgang
(4) Last

Diagnose: Überlast

19437
15249

Überlast kann nur an einem Ausgang mit Strommessung erkannt werden.

Überlast ist definiert als ...
"nominaler Maximalstrom laut Datenblatt + 12,5 %".

Diagnose: Leiterbruch (via Strommessung)

19435
19400

Eine Leiterbruch-Erkennung erfolgt über den Rücklesekanal. Bei geschaltetem Ausgang (Qn=TRUE) wird dann ein Leiterbruch erkannt, wenn über den Widerstand R_i kein Strom fließt (keine Spannung abfällt). Ohne den Leiterbruch fließt durch den Längswiderstand R_i der Laststrom und erzeugt damit einen Spannungsabfall, der über den Rücklesekanal ausgewertet wird.

© ifm electronic gmbh www.ifm.com

Diagnose: Leiterbruch (via Spannungsmessung)

19436
19404

Eine Leiterbruch-Erkennung erfolgt über den Rücklesekanal. Bei gesperrtem Ausgang (Qn=FALSE) wird dann ein Leiterbruch erkannt, wenn der Widerstand R_u den Rücklesekanal auf HIGH-Potential (VBB) zieht. Ohne den Leiterbruch würde die niederohmige Last (R_L < 10 kOhm) LOW (logisch 0) erzwingen.

Diagnose: Kurzschluss

19405

Eine Kurzschluss-Erkennung erfolgt über den Rücklesekanal. Bei geschaltetem Ausgang (Qn=TRUE) wird dann ein Kurzschluss gegen GND erkannt, wenn der Rücklesekanal auf LOW-Potential (GND) gezogen wird.

© ifm electronic gmbh www.ifm.com

3.2.4 Hinweise zur Anschlussbelegung

1426

Die Anschlussbelegungen (→ Montageanleitungen der Geräte, Kapitel "Anschlussbelegung") beschreiben die Standard-Gerätekonfigurationen. Die Anschlussbelegung dient der Zuordnung der Ein- und Ausgangskanäle zu den IEC-Adressen und den Geräteanschlussklemmen.

Die einzelnen Kürzel haben folgende Bedeutung:

A	Analog-Eingang
BH	Binärer highside-Eingang: minus-schaltend für negatives Sensorsignal Binärer highside-Ausgang: plus-schaltend für positives Ausgangssignal
BL	Binärer lowside-Eingang: plus-schaltend für positives Sensorsignal Binärer lowside-Ausgang: minus-schaltend für negatives Ausgangssignal
CYL	Eingang Periodendauermessung
ENC	Eingang Drehgebersignale
FRQ	Frequenzeingang
H-Bridge	Ausgang mit H-Brücken-Funktion
PWM	**P**uls**w**eiten-**m**oduliertes Signal
PWMi	PWM-Ausgang mit Strommessung
IH	Impuls-/Zählereingang, highside, minus-schaltend für negatives Sensorsignal
IL	Impuls-/Zählereingang, lowside, plus-schaltend für positives Sensorsignal
R	Rücklesekanal für einen Ausgang

Zuordnung der Ein-/Ausgangskanäle: → Katalog, Montageanleitung oder Datenblatt

3.2.5 Sicherheitshinweise zu Reed-Relais

7348

Beim Einsatz von nichtelektronischen Schaltern Folgendes beachten:

Kontakte von Reed-Relais können (reversibel) verkleben, wenn sie ohne Vorwiderstand an den Geräte-Eingängen angeschlossen werden.

- **Abhilfe:** Vorwiderstand zum Reed-Relais installieren:
  Vorwiderstand = max. Eingangsspannung / zulässiger Strom im Reed-Relais
  Beispiel: 32 V / 500 mA = 64 Ohm
- Der Vorwiderstand darf 5 % des Eingangswiderstands RE des Geräte-Eingangs (→ Datenblatt) nicht überschreiten. Sonst wird das Signal nicht als TRUE erkannt.
  Beispiel:
  RE = 3 000 Ohm
  ⇨ max. Vorwiderstand = 150 Ohm

© ifm electronic gmbh www.ifm.com

3.2.6 Status-LED

7998

Die Betriebszustände werden durch die integrierte Status-LED (Voreinstellung) angezeigt.

LED-Farbe	Blinkfrequenz	Beschreibung
aus	konstant aus	keine Betriebsspannung
Orange	kurzzeitig ein	INIT-Zustand, Reset-Checks
Grün	5 Hz	kein Laufzeitsystem geladen
Grün	2 Hz	RUN-Zustand: Anwendungsprogramm läuft
Grün	konstant ein	STOP-Zustand: Anwendungsprogramm angehalten
Rot	5 Hz	STOP-Zustand mit Fehler: Anwendungsprogramm angehalten Ursache: Unterspannung
Rot	10 Hz	STOP-Zustand mit Fehler: Anwendungsprogramm angehalten Ursache: Laufzeitüberschreitung des Anwendungsprogramms oder der Visualisierung: ▶ Anwendungsprogramm löschen! ▶ PowerOn-Reset ▶ Anwendungsprogramm neu ins Gerät laden
Rot	konstant ein	FATAL-ERROR: Anwendungsprogramm angehalten Ursache: Software-Watchdog ist ausgefallen ▶ PowerOn-Reset Wenn ohne Erfolg: ▶ Goto Bootloader ▶ PowerOn-Reset ▶ BasicSystem neu ins Gerät laden ▶ Anwendungsprogramm neu ins Gerät laden Wenn ohne Erfolg: ▶ Hardware-Fehler: Gerät an ifm einsenden!

Die Betriebszustände STOP und RUN können vom Programmiersystem geändert werden.

LED im Anwendungsprogramm steuern

15481

Mit ***SET_LED*** (→ Seite 208) können im Anwendungsprogramm für den RUN-Zustand Frequenz und Farbe der Status-LED geändert werden.

Der Einsatz des LED-Bausteins im Anwendungsprogramm ersetzt im RUN-Zustand die System-Voreinstellung der Status-LED.

© ifm electronic gmbh www.ifm.com

3.3 Schnittstellen-Beschreibung

Inhalt

CAN-Schnittstellen 26

14098

3.3.1 CAN-Schnittstellen

Inhalt

CAN: Schnittstellen und Protokolle 26

14101

Anschlüsse und Daten → Datenblatt

CAN: Schnittstellen und Protokolle

14589
15238

Die Geräte werden je nach Aufbau der Hardware mit mehreren CAN-Schnittstellen ausgerüstet. Grundsätzlich können alle Schnittstellen unabhängig voneinander mit folgenden Funktionen genutzt werden:

- RAW-CAN (Layer 2): CAN auf Ebene 2 (→ Kapitel ***Bausteine: RAW-CAN (Layer 2)*** (→ Seite 67))
- CANopen-Master / CANopen-Slave (→ Kapitel ***Bausteine: CANopen*** (→ Seite 93))
- CANopen-Netzwerkvariablen (via CODESYS) (→ Kapitel ***Netzwerkvariablen*** (→ Seite 58))
- SAE J1939 (für Antriebsmanagement, → Kapitel ***Bausteine: SAE J1939*** (→ Seite 138))
- Buslast-Erkennung
- Errorframe-Zähler
- Download-Schnittstelle
- 100 % Buslast ohne Paketverlust

14591

In diesem ecomat*mobile*-Gerät sind folgende CAN-Schnittstellen und CAN-Protokolle verfügbar:

CAN-Schnittstelle	CAN 1	CAN 2	CAN 3	CAN 4
voreingestellte Download-ID	ID 127	ID 126	ID 125	ID 124
CAN-Protokolle	CAN Layer 2	CAN Layer 2	Schnittstelle nicht vorhanden	Schnittstelle nicht vorhanden
	CANopen	CANopen		
	SAE J1939	SAE J1939		

Standard-Baudrate = 250 kBit/s

Alle CAN-Schnittstellen können zeitgleich mit allen CAN-Protokollen arbeiten. Die verwendeten IDs dürfen sich nicht beeinträchtigen!

© ifm electronic gmbh

3.4 Software

Inhalt

Software-Module für das Gerät 27
Programmierhinweise für CODESYS-Projekte 30
Betriebszustände 34
Leistungsgrenzen des Geräts 37

14107

3.4.1 Software-Module für das Gerät

Inhalt

Bootloader 28
Laufzeitsystem 28
Anwendungsprogramm 28
Bibliotheken 29

14110

Die Software in diesem Gerät setzt wie folgt auf der Hardware auf:

Software-Modul	Anwender kann das Modul ändern?	womit?
Anwendungsprogramm mit Bibliotheken	ja	CODESYS, MaintenanceTool
Laufzeitsystem (LZS) *)	Upgrade ja Downgrade ja	MaintenanceTool
Bootloader	nein	---
(Hardware)	nein	---

*) Die Laufzeitsystem-Versionsnummer muss der Target-Versionsnummer in der CODESYS-Zielsystemeinstellung entsprechen!
→ Kapitel *Target einrichten* (→ Seite 43)

Nachfolgend beschreiben wir diese Software-Module:

© ifm electronic gmbh

Bootloader

14111

Im Auslieferungszustand enthalten ecomatmobile-Controller nur den Bootloader.
Der Bootloader ist ein Startprogramm, mit dem das Laufzeitsystem und das Anwendungsprogramm auf dem Gerät nachgeladen werden können.
Der Bootloader enthält Grundroutinen...
- zur Kommunikation der Hardware-Module untereinander,
- zum Nachladen des Laufzeitsystems.

Der Bootloader ist das erste Software-Modul, das im Gerät gespeichert sein muss.

Laufzeitsystem

14112

Grundprogramm im Gerät, stellt die Verbindung her zwischen der Hardware des Gerätes und dem Anwendungsprogramm.
→ Kapitel *Software-Module für das Gerät* (→ Seite 27)

Im Auslieferungszustand ist im Normalfall kein Laufzeitsystem im Controller geladen (LED blinkt grün mit 5 Hz). In diesem Betriebszustand ist nur der Bootloader aktiv. Dieser stellt die minimalen Funktionen für den Laufzeitsystem-Ladevorgang zur Verfügung, u.a. die Unterstützung der Schnittstellen (z.B. CAN).

Der Laufzeitsystem-Download muss im Normalfall nur einmalig durchgeführt werden. Das Anwendungsprogramm kann anschließend (auch mehrmals) in den Controller geladen werden, ohne das Laufzeitsystem zu beeinflussen.

Das Laufzeitsystem wird zusammen mit dieser Dokumentation auf einem separaten Datenträger zur Verfügung gestellt. Zusätzlich kann auch die aktuelle Version von der Homepage der ifm electronic gmbh heruntergeladen werden:
→ www.ifm.com > Land wählen > [Service] > [Download]

Anwendungsprogramm

14118

Software, die speziell für die Anwendung vom Hersteller in die Maschine programmiert wird. Die Software enthält üblicherweise logische Sequenzen, Grenzwerte und Ausdrücke zum Steuern der entsprechenden Ein- und Ausgänge, Berechnungen und Entscheidungen.

8340

⚠ WARNUNG

Für die sichere Funktion der Anwendungsprogramme, die vom Anwender erstellt werden, ist dieser selbst verantwortlich. Bei Bedarf muss er zusätzlich entsprechend der nationalen Vorschriften eine Abnahme durch entsprechende Prüf- und Überwachungsorganisationen durchführen lassen.

© ifm electronic gmbh www.ifm.com

Bibliotheken

15409

ifm electronic bietet passend für jedes Gerät eine Reihe von Bibliotheken (`*.LIB`) an, die Programmmodule für das Anwendungsprogramm enthalten. Beispiele:

Bibliothek	Verwendung
`ifm_CR0411_Vxxyyzz.LIB`	gerätespezifische Bibliothek Muss immer im Anwendungsprogramm enthalten sein!
`ifm_RawCAN_NT_Vxxyyzz.LIB`	(optional) wenn eine CAN-Schnittstelle des Geräts mit CAN Layer 2 betrieben werden soll
`ifm_CANopen_NT_Vxxyyzz.LIB`	(optional) wenn eine CAN-Schnittstelle des Geräts als CANopen-Master oder als CANopen-Slave betrieben werden soll
`ifm_J1939_NT_Vxxyyzz.LIB`	(optional) wenn eine CAN-Schnittstelle des Geräts mit einer Motorsteuerung kommunizieren soll

Detailinformationen: → *ifm-Bibliotheken für das Gerät CR0411* (→ Seite 59)

© ifm electronic gmbh

3.4.2 Programmierhinweise für CODESYS-Projekte

Inhalt

FB, FUN, PRG in CODESYS 30
Zykluszeit beachten! 31
Anwendungsprogramm erstellen 32
ifm-Maintenance-Tool nutzen 33
Verteilen des Anwendungsprogramms 33

7426

Hier erhalten Sie Tipps zum Programmieren des Geräts.

► Beachten Sie die Hinweise im CODESYS-Programmierhandbuch
→ www.ifm.com > Land wählen > [Datenblattsuche] > CR0411 > [Betriebsanleitungen],
→ ecomat*mobile*-DVD "Software, tools and documentation".

FB, FUN, PRG in CODESYS

15410

In CODESYS unterscheiden wir folgende Typen von Bausteinen (POUs):

FB = function block = Funktionsbaustein
- Ein FB kann mehrere Eingänge und mehrere Ausgänge haben.
- Ein FB darf in einem Projekt mehrmals aufgerufen werden.
- Für jeden Aufruf muss eine Instanz deklariert werden.
- Erlaubt: Im FB aufrufen von FB und FUN.

FUN = function = Funktion
- Eine Funktion kann mehrere Eingänge, aber nur einen Ausgang haben.
- Der Ausgang ist vom gleichen Datentyp wie die Funktion selbst.

PRG = program = Programm
- Ein PRG kann mehrere Eingänge und mehrere Ausgänge haben.
- Ein PRG darf in einem Projekt nur einmal aufgerufen werden.
- Erlaubt: im PRG aufrufen von PRG, FB und FUN.

HINWEIS

Funktionsbausteine dürfen NICHT in Funktionen aufgerufen werden!
Sonst: Bei der Ausführung stürzt das Anwendungsprogramm ab.

Alle Bausteine (POUs) dürfen NICHT rekursiv aufgerufen werden, auch nicht indirekt!

Eine IEC-Anwendung darf maximal 8000 Bausteine (POUs) enthalten, in diesem Gerät maximal 512 Bausteine (POUs)!

Hintergrund:

Alle Variablen von Funktionen...
- werden beim Aufruf initialisiert und
- werden nach der Rückkehr zum Aufrufer ungültig.

Funktionsbausteine haben 2 Aufrufe:
- einen Initialisierungsaufruf und
- den eigentlichen Aufruf, um irgend etwas zu tun.

Folglich heißt das für den FB-Aufruf in einer Funktion:
- jedesmal erfolgt ein zusätzlicher Initialisierungsaufruf und
- die Daten des letzten Aufrufs gehen verloren.

© ifm electronic gmbh

Zykluszeit beachten!

8006

Bei den frei programmierbaren Geräten aus der Controller-Familie **ecomatmobile** stehen in einem großen Umfang Bausteine zur Verfügung, die den Einsatz der Geräte in den unterschiedlichsten Anwendungen ermöglichen.

Da diese Bausteine je nach Komplexität mehr oder weniger Systemressourcen belegen, können nicht immer alle Bausteine gleichzeitig und mehrfach eingesetzt werden.

ACHTUNG

Gefahr von zu trägem Verhalten des Geräts!
Zykluszeit darf nicht zu lang werden!

- ► Beim Erstellen des Anwendungsprogramms die oben aufgeführten Empfehlungen beachten und durch Austesten überprüfen.
- ► Bei Bedarf durch Neustrukturieren der Software und des Systemaufbaus die Zykluszeit vermindern.

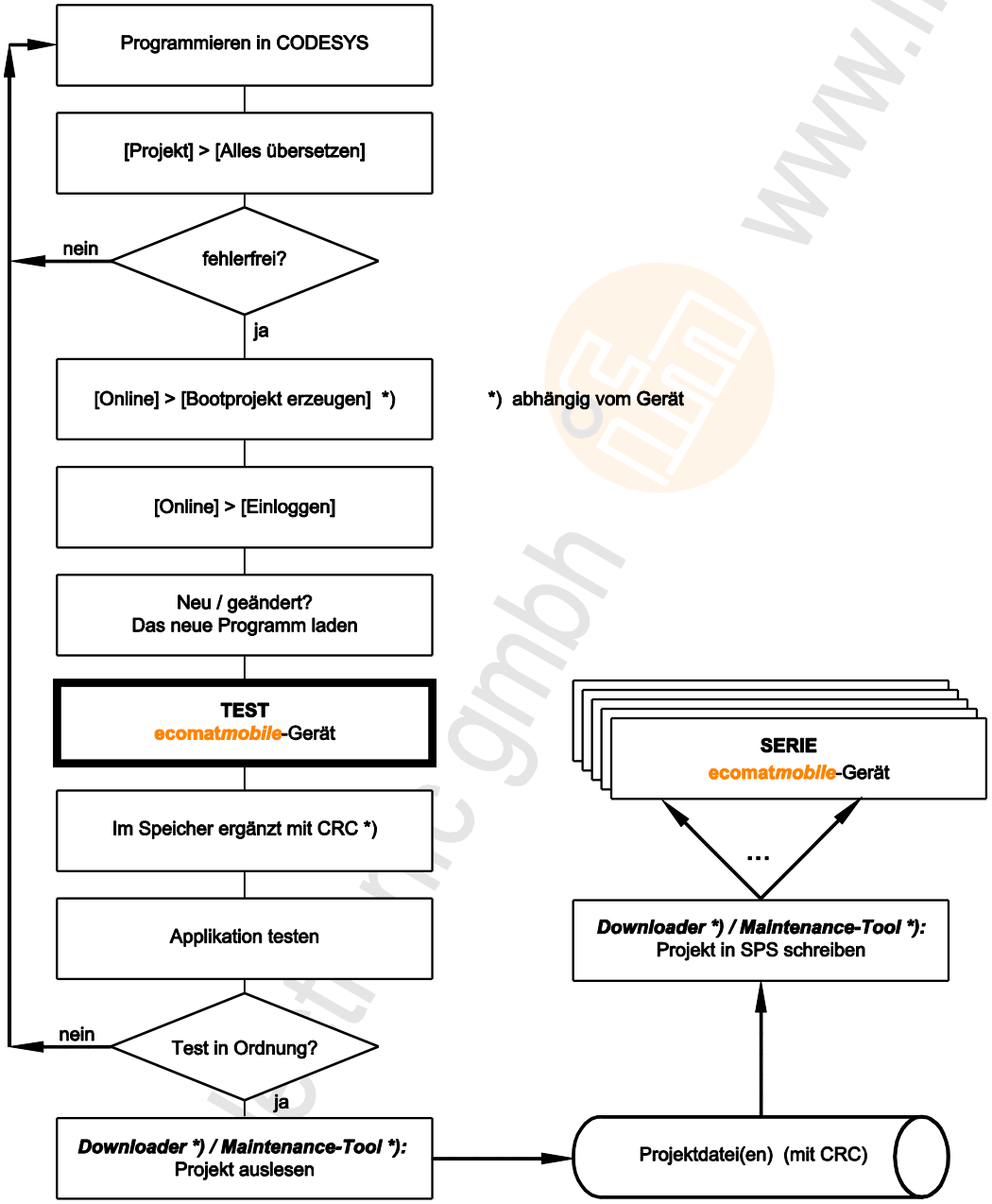
© ifm electronic gmbh www.ifm.com

Anwendungsprogramm erstellen

8007

Das Anwendungsprogramm wird mit dem Programmiersystem CODESYS erstellt und während der Programmentwicklung mehrfach zum Testen in die Steuerung geladen:
In CODESYS: [Online] > [Einloggen] > das neue Programm laden.

Für jeden derartigen Download via CODESYS wird dazu der Quellcode neu übersetzt. Daraus resultiert, dass auch jedes Mal im Speicher der Steuerung eine neue Prüfsumme gebildet wird. Auch für Sicherheitssteuerungen ist dieses Verfahren bis zur Freigabe der Software zulässig.



Grafik: Erstellen und Verteilen der Software

© ifm electronic gmbh www.ifm.com

ifm-Maintenance-Tool nutzen

8492

Das **ifm**-Maintenance-Tool dient dem einfachen Übertragen des Programmcodes vom Programmierplatz in das Gerät. Grundsätzlich kann jedes Anwendungsprogramm mit dem **ifm**-Maintenance-Tool auf die Geräte kopiert werden. Vorteil: Dazu ist kein Programmiersystem mit einer CODESYS-Lizenz erforderlich.

Hier finden Sie das aktuelle **ifm**-Maintenance-Tool:
→ www.ifm.com > Land wählen > [Service] > [Download] > [Systeme für mobile Arbeitsmaschinen]
→ **ecomat*mobile***-DVD "Software, tools and documentation" im Register "R360 tools [D/E]"

Verteilen des Anwendungsprogramms

8493

Wir empfehlen folgenden Ablauf, wenn das Anwendungsprogramm auf Serienmaschinen kopiert wird und zum Einsatz kommt:

- Sichern der Software
  Nach Abschluss der Programmentwicklung muss die letzte Version des in das Gerät geladenen Anwendungsprogramms mit dem **ifm**-Maintenance-Tool zunächst aus dem Gerät ausgelesen und auf einem Datenträger unter dem Namen `projektdatei.RESX` gespeichert werden. Nur dieses Verfahren gewährleistet, dass das Anwendungsprogramm mit den entsprechenden Prüfsummen gesichert ist.
- Download der Software
  Um in der Serienproduktion alle Maschinen mit einer einheitlichen Software auszurüsten, darf nur diese Datei mit dem **ifm**-Maintenance-Tool in die Geräte geladen werden.
- Ein Fehler in den Daten dieser Datei wird durch die integrierte Prüfsumme beim erneuten Laden durch das **ifm**-Maintenance-Tool automatisch erkannt.

© ifm electronic gmbh www.ifm.com

3.4.3 Betriebszustände

1075

Nach Anlegen der Versorgungsspannung kann sich das ecomat*mobile*-Gerät in einem von fünf möglichen Betriebszuständen befinden:
- BOOTLOADER
- INIT
- STOP
- RUN
- SYSTEM STOP

INIT-Zustand (Reset)

1076

Voraussetzung: ein gültiges Laufzeitsystem ist installiert.

Dieser Zustand wird nach jedem Power-On-Reset durchlaufen:

> Das Laufzeitsystem wird initialisiert.

> Verschiedene Checks werden durchgeführt, z.B. Warten auf gültige Versorgungsspannung.

> Dieser nur temporäre Zustand wird vom RUN- oder STOP-Zustand abgelöst.

> Die LED leuchtet orange.

Wechsel aus diesem Zustand in einen der folgenden Zustände möglich:
- RUN
- STOP

STOP-Zustand

8288

Wechsel in diesen Zustand in folgenden Fällen möglich:
- Aus dem INIT-Zustand, wenn kein Anwendungsprogramm geladen ist.
- Aus dem RUN-Zustand, wenn folgende Bedingung erfüllt ist:
  - STOP-Kommando kommt über die CODESYS-Schnittstelle.

Im STOP-Zustand:

> Die Ausgänge des Geräts sind abgeschaltet.

> Das Abarbeiten des Anwendungsprogramms ist angehalten.

> Die LED leuchtet grün.

Wechsel aus diesem Zustand in einen der folgenden Zustände möglich:
- RUN
- ERROR
- FATAL ERROR
- INIT (nach Power-On-Reset)

© ifm electronic gmbh www.ifm.com

RUN-Zustand

8287

Wechsel in diesen Zustand in folgenden Fällen möglich:

- Aus dem INIT-Zustand (Autostart), wenn folgende Bedingungen erfüllt sind:
  - Die Betriebsspannung hat einen Mindestwert erreicht. UND:
  - Das Anwendungsprogramm ist vorhanden.
- Aus dem STOP-Zustand:
  - durch das CODESYS-RUN-Kommando.
  - Die Betriebsspannung hat einen Mindestwert erreicht oder überschritten.

Im RUN-Zustand:

> Das Laufzeitsystem läuft.

> Das Anwendungsprogramm läuft.

> Die LED blinkt grün mit 2 Hz.
Die LED kann vom Anwendungsprogramm abweichend gesteuert werden → FB ***SET_LED*** (→ Seite 208).

Wechsel aus diesem Zustand in einen der folgenden Zustände möglich:
- INIT (nach Power-On-Reset)
- STOP
- ERROR
- FATAL ERROR

ERROR-Zustand

8290

Wechsel in diesen Zustand in folgenden Fällen möglich:
- bei zu geringer Versorgungsspannung.

Im ERROR-Zustand:

> Die Ausgänge des Geräts sind ausgeschaltet.

> Das Abarbeiten des Anwendungsprogramms ist angehalten.

> Systemparameter werden gespeichert.

> Die LED blinkt rot mit 5 Hz.

Wechsel aus diesem Zustand in einen der folgenden Zustände möglich:
- RUN
- STOP
- FATAL ERROR
- INIT (nach Power-On-Reset)

© ifm electronic gmbh www.ifm.com

FATAL-ERROR-Zustand

8289

Wechsel in diesen Zustand in folgenden Fällen möglich:
- Speicherfehler (RAM / Flash)
- Ausnahmezustand
- Laufzeitsystem-Fehler

Im FATAL-ERROR-Zustand:

> Die Ausgänge des Geräts sind abgeschaltet.

> Das Anwendungsprogramm ist beendet.

> Das Laufzeitsystem ist beendet.

> Die LED leuchtet rot.

Wechsel aus diesem Zustand in einen der folgenden Zustände möglich:
- INIT (nach Power-On-Reset)

© ifm electronic gmbh www.ifm.com

3.4.4 Leistungsgrenzen des Geräts

7358

Leistungsgrenzen des Geräts beachten! → Datenblatt

Verhalten des Watchdog

15365

Ein Watchdog überwacht in diesem Gerät die Programmlaufzeit der CODESYS-Anwendung.

Wird die maximale Watchdog-Zeit (100 ms) überschritten:
> Gerät geht in den Zustand "Timeout Error"
> alle Prozesse werden angehalten (Reset)
> alle Ausgänge werden ausgeschaltet
> die Status-LED blinkt rot mit 10 Hz

Störung beseitigen:
- ▶ Anwendungsprogramm löschen!
- ▶ PowerOn-Reset
- ▶ Anwendungsprogramm neu ins Gerät laden

Sollte der beschriebene Watchdog ausfallen, dann:
> ein zweiter Watchdog führt das Gerät in den Zustand "Fatal Error"
> die Status-LED leuchtet rot

Störung beseitigen:
- ▶ PowerOn-Reset

Wenn ohne Erfolg:
- ▶ Goto Bootloader
- ▶ PowerOn-Reset
- ▶ Laufzeitsystem neu ins Gerät laden
- ▶ Anwendungsprogramm neu ins Gerät laden

Wenn ohne Erfolg:
- ▶ Hardware-Fehler: Gerät an ifm einsenden!

© ifm electronic gmbh www.ifm.com

Grenzen für CAN in diesem Gerät

17975

FIFO (**F**irst **I**n, **F**irst **O**ut) = Arbeitsweise des Stapelspeichers: Das Datenpaket, das zuerst in den Stapelspeicher geschrieben wurde, wird auch als erstes gelesen. Pro Identifier steht ein solcher Zwischenspeicher (als Warteschlange) zur Verfügung.

Einige RAW-CAN-Bausteine ermöglichen das Versenden und Empfangen mehrerer Nachrichten innerhalb eines SPS-Zyklus, da die Nachrichten in einem FiFo zwischengespeichert werden:
- CAN_TX..., → Bausteine: RAW-CAN Daten senden
- ***CAN_RX_ENH_FIFO*** (→ Seite 77)
- ***CAN_RX_RANGE_FIFO*** (→ Seite 81)

Die Anzahl der FiFo-Nachrichten ist begrenzt. Es gelten folgende Leistungsgrenzen der Geräte:

Gerät / **Kriterium**	BasicController: CR040n, CR041n, CR043n BasicDisplay: CR045n SmartController: CR253n	PDM360 NG: CR108n
max. FiFo senden - mit FB CAN_TX... - mit FB CAN_TX_ENH...	 4 Nachrichten 16 Nachrichten	 4 Nachrichten 16 Nachrichten
max. FiFo empfangen - mit FB CAN_RX_..._FIFO	32 Nachrichten	32 Nachrichten

Grenzen für CANopen in diesem Gerät

17976

Es gelten folgende Leistungsgrenzen der Geräte:

Gerät / **Kriterium**	BasicController: CR040n, CR041n, CR043n BasicDisplay: CR045n SmartController: CR253n	PDM360 NG: CR108n
max. Guarding-Fehler	32 Meldungen	128 Meldungen
max. SDO-Daten	2 048 Bytes	2 048 Bytes

Grenzen für CAN J1939 in diesem Gerät

17977

Es gelten folgende Leistungsgrenzen der Geräte:

Gerät / **Kriterium**	BasicController: CR040n, CR041n, CR043n BasicDisplay: CR045n SmartController: CR253n	PDM360 NG: CR108n
max. FiFo senden - mit FB J1939_TX - mit FB J1939_TX_ENH	 4 Nachrichten 16 Nachrichten	 4 Nachrichten 16 Nachrichten
max. FiFo empfangen - mit FB J1939_RX_FIFO	32 Nachrichten	32 Nachrichten
max. DTCs	64 Meldungen	64 Meldungen
max. Daten J1939	1 785 Bytes	1 785 Bytes

© ifm electronic gmbh www.ifm.com

4 Konfigurationen

Inhalt

Laufzeitsystem einrichten ... 39
Programmiersystem einrichten ... 42
Funktionskonfiguration, allgemein ... 46
Funktionskonfiguration der Ein- und Ausgänge ... 47
Variablen ... 56

Die in den jeweiligen Montage- und Installationsanweisungen oder dem ***Anhang*** (→ Seite 214) dieser Dokumentation beschriebenen Gerätekonfigurationen stehen als Standardgeräte (Lagerware) zur Verfügung. Diese decken bei den meisten Anwendungen die geforderten Spezifikationen ab.

Entsprechend den Kundenanforderungen bei Serieneinsatz ist es aber auch möglich, dass andere Gerätekonfigurationen z.B. hinsichtlich der Zusammenstellung der Ein- und Ausgänge und der Ausführung der Analogkanäle eingesetzt werden.

4.1 Laufzeitsystem einrichten

Inhalt

Laufzeitsystem neu installieren ... 40
Laufzeitsystem aktualisieren ... 41
Installation verifizieren ... 41

14091

© ifm electronic gmbh

4.1.1 Laufzeitsystem neu installieren

14635
8486

Im Auslieferungszustand ist im Normalfall kein Laufzeitsystem im Gerät geladen (LED blinkt grün mit 5 Hz). In diesem Betriebszustand ist nur der Bootloader aktiv. Dieser stellt die minimalen Funktionen für den Laufzeitsystem-Ladevorgang zur Verfügung, u.a. die Unterstützung der Schnittstellen (z.B. RS232, CAN).

Der Laufzeitsystem-Download muss im Normalfall nur einmalig durchgeführt werden. Das Anwendungsprogramm kann anschließend (auch mehrmals) in das Gerät geladen werden, ohne das Laufzeitsystem zu beeinflussen.

Das Laufzeitsystem wird zusammen mit dieser Dokumentation auf einem separaten Datenträger zur Verfügung gestellt. Zusätzlich kann auch die aktuelle Version von der Homepage der **ifm electronic gmbh** heruntergeladen werden:
→ www.ifm.com > Land wählen > [Service] > [Download] > [Systeme für mobile Arbeitsmaschinen]

ACHTUNG

Gefahr von Datenverlust!

Bei Spannungsausfall während der Datenübertragung können Daten verloren gehen, so dass das Gerät nicht mehr funktionsfähig ist. Reparatur ist nur bei **ifm electronic** möglich.

► Für ständige Spannungsversorgung während der Datenübertragung sorgen!

HINWEIS

Es müssen immer die zum gewählten Target passenden Software-Stände zum Einsatz kommen:
- des Laufzeitsystems (`ifm_CR0411_Vxxyyzz.RESX`),
- der Steuerungskonfiguration (`ifm_CR0411_Vxx.CFG`),
- der Gerätebibliothek (`ifm_CR0411_Vxxyyzz.LIB`) und
- der weiteren Dateien

V	Version
xx: 00...99	Versionsnummer
yy: 00...99	Release-Nummer
zz: 00...99	Patch-Nummer

Dabei müssen der Basisdateiname (z.B. "`CR0411`") und die Software-Versionsnummer "xx" (z.B. "`01`") überall den gleichen Wert haben! Andernfalls geht das Gerät in den STOP-Zustand

Die Werte für "yy" (Release-Nummer) und "zz" (Patch-Nummer) müssen **nicht** übereinstimmen.

4368

Folgende Dateien müssen ebenfalls geladen sein:
- die zum Projekt erforderlichen internen Bibliotheken (in IEC 61131 erstellt),
- die Konfigurationsdateien (`*.CFG`)
- und die Target-Dateien (`*.TRG`).

Es kann vorkommen, dass das Zielsystem mit Ihrer aktuell installierten Version von CODESYS nicht oder nur teilweise programmiert werden kann. Im diesem Fall wenden Sie sich bitte an den technischen Support der **ifm electronic gmbh**.

Das Laufzeitsystem wird mit dem eigenständigen Programm "Maintenance Tool" in das Gerät übertragen. (Das Programm befindet sich auf der **ecomat*mobile***-DVD "Software, tools and documentation" oder kann bei Bedarf von der **ifm**-Homepage heruntergeladen werden:
→ www.ifm.com > Land wählen > [Service] > [Download] > [Systeme für mobile Arbeitsmaschinen])

Das Anwendungsprogramm wird im Normalfall über das Programmiersystem in das Gerät geladen. Es kann aber ebenfalls mit dem "Maintenance Tool" geladen werden, wenn es zuvor aus dem Gerät ausgelesen wurde.

© ifm electronic gmbh www.ifm.com

4.1.2 Laufzeitsystem aktualisieren

13269

Auf dem Gerät ist bereits ein älteres Laufzeitsystem installiert. Nun möchten Sie das Laufzeitsystem auf dem Gerät aktualisieren?

14158

ACHTUNG

Gefahr von Datenverlust!

Beim Löschen oder Aktualisieren des Laufzeitsystems werden alle Daten und Programme auf dem Gerät gelöscht.

► Alle erforderlichen Daten und Programme sichern, bevor das Laufzeitsystem gelöscht oder aktualisiert wird!

3084

Immer, wenn es zu wesentlichen Verbesserungen in der Betriebsystem-Software oder des CODESYS-Laufzeitsystems kommt, gibt ifm davon eine neue Version heraus. Die Versionen werden fortlaufend durchnummeriert (V01, V02, V03, ...).

Welche neuen Zusatzfunktionen die neue Softwareversion enthält, entnehmen Sie bitte der jeweiligen Dokumentation. Beachten Sie, ob in der Dokumentation auf besondere Anforderungen an die Hardware-Version hingewiesen wird.

Wenn Sie im Besitz eines Gerätes mit einer älteren Version sind und wenn die Bedingungen für die Hardware und Ihr Projekt stimmen, können Sie Ihr Gerät durch Aktualisieren der Software auf den neuen Software-Stand bringen.

Prinzipiell gelten für diesen Vorgang die gleichen Hinweise, wie zuvor im Kapitel 'Laufzeitsystem neu installieren' gegeben wurden.

4.1.3 Installation verifizieren

14637

► Nach dem Laden des Laufzeitsystems in die Steuerung:
  • Prüfen, ob das Laufzeitsystem korrekt übertragen wurde!
  • Prüfen, ob sich das richtige Laufzeitsystem auf der Steuerung befindet!

► 1. Prüfung:
mit dem ifm-Maintenance-Tool prüfen, ob die richtige Laufzeitsystem-Version geladen wurde:
  • Name und Version des Laufzeitsystems im Gerät auslesen!
  • Diese Daten manuell mit den Soll-Daten vergleichen!

► 2. Prüfung (optional):
Im Anwendungsprogramm prüfen, ob die richtige Laufzeitsystem-Version geladen wurde:
  • Name und die Version des Laufzeitsystems im Gerät auslesen!
  • Diese Daten mit fest vorgegebenen Werten vergleichen!
Zum Auslesen der Daten dient folgender FB:

GET_SW_INFO (→ Seite 200)	liefert Informationen über die System-Software des Geräts: • Software-Name, • Software-Version, • Build-Nummer, • Build-Datum

© ifm electronic gmbh www.ifm.com

4.2 Programmiersystem einrichten

Inhalt

Programmiersystem manuell einrichten 42
Programmiersystem über Templates einrichten 46

14461

4.2.1 Programmiersystem manuell einrichten

Inhalt

Target einrichten 43
Steuerungskonfiguration aktivieren 44
CAN-Schnittstellen deklarieren (z.B. CR1080) 45

3963

© ifm electronic gmbh

Target einrichten

13136
11379

Beim Erstellen eines neuen Projektes in CODESYS muss die dem Gerät entsprechende Target-Datei geladen werden.

► Im Dialog-Fenster [Zielsystem Einstellungen] im Menü [Konfiguration] die gewünschte Target-Datei wählen.

> Die Target-Datei stellt für das Programmiersystem die Schnittstelle zur Hardware her.

> Gleichzeitig mit Wahl des Targets werden automatisch einige wichtige Bibliotheken und die Steuerungskonfiguration geladen.

► Bei Bedarf im Fenster [Zielsystem Einstellungen] > Reiter [Netzfunktionen] > [Parameter-Manager unterstützen] und / oder [Netzvariablen unterstützen] aktivieren.

► Bei Bedarf geladene (3S-)Bibliotheken wieder entfernen oder durch weitere (ifm-)Bibliotheken ergänzen.

► Immer die passende Geräte-Bibliothek `ifm_CR0411_Vxxyyzz.LIB` manuell ergänzen!

HINWEIS

Es müssen immer die zum gewählten Target passenden Software-Stände zum Einsatz kommen:
- des Laufzeitsystems (`ifm_CR0411_Vxxyyzz.RESX`),
- der Steuerungskonfiguration (`ifm_CR0411_Vxx.CFG`),
- der Gerätebibliothek (`ifm_CR0411_Vxxyyzz.LIB`) und
- der weiteren Dateien

V	Version
xx: 00...99	Versionsnummer
yy: 00...99	Release-Nummer
zz: 00...99	Patch-Nummer

Dabei müssen der Basisdateiname (z.B. "`CR0411`") und die Software-Versionsnummer "xx" (z.B. "`01`") überall den gleichen Wert haben! Andernfalls geht das Gerät in den STOP-Zustand

Die Werte für "yy" (Release-Nummer) und "zz" (Patch-Nummer) müssen **nicht** übereinstimmen.

4368

Folgende Dateien müssen ebenfalls geladen sein:
- die zum Projekt erforderlichen internen Bibliotheken (in IEC 61131 erstellt),
- die Konfigurationsdateien (`*.CFG`)
- und die Target-Dateien (`*.TRG`).

Es kann vorkommen, dass das Zielsystem mit Ihrer aktuell installierten Version von CODESYS nicht oder nur teilweise programmiert werden kann. Im diesem Fall wenden Sie sich bitte an den technischen Support der **ifm electronic gmbh**.

© ifm electronic gmbh www.ifm.com

Steuerungskonfiguration aktivieren

10079

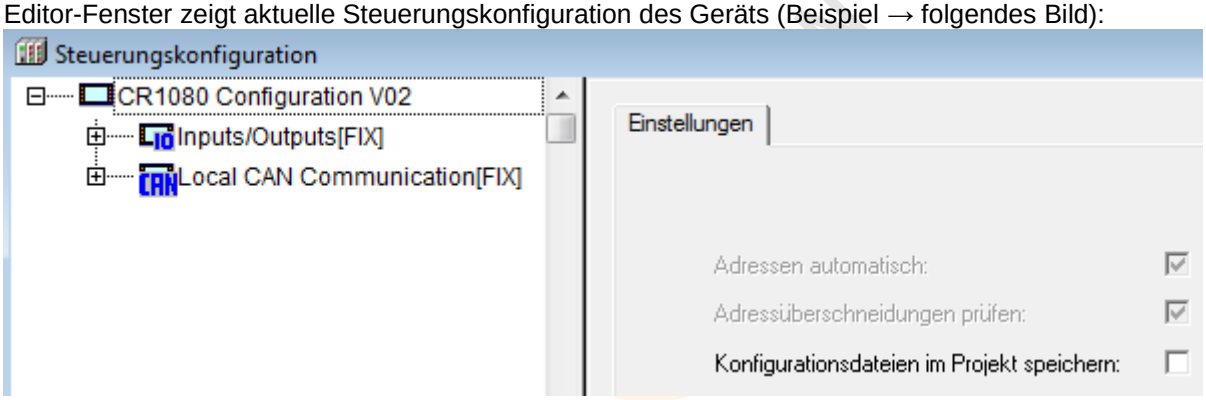
Die Steuerungskonfiguration wird automatisch zusammen mit dem Zielsystem geladen. Die Steuerungskonfiguration bildet den Inhalt der Datei CR0411.cfg in CODESYS ab. Der Programmierer hat dadurch einfachen Zugriff auf vordefinierte System- und Fehlermerker, Ein- und Ausgänge sowie die CAN-Schnittstellen des Geräts.

Um auf die Steuerungskonfiguration zuzugreifen (Bsp.: CR1080):

► In CODESYS den Reiter [Ressourcen] wählen:

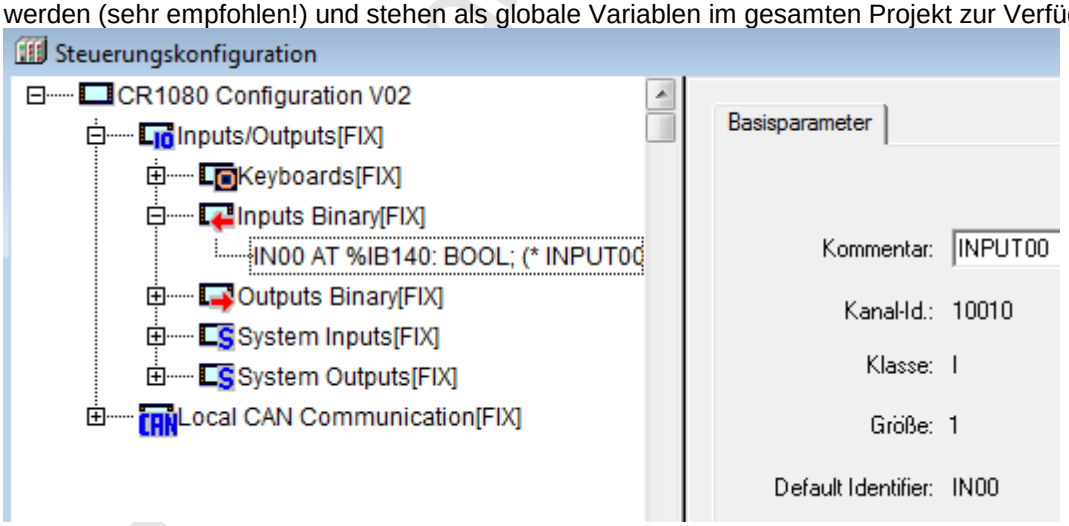
► Im Ressourcen-Baum: Doppelklick auf [Steuerungskonfiguration]

> Editor-Fenster zeigt aktuelle Steuerungskonfiguration des Geräts (Beispiel → folgendes Bild):



Der Programmierer hat über die Steuerungskonfiguration einfachen Zugriff auf folgende Komponenten:

- System- und Fehlermerker
  Je nach Anwendung und Anwendungsprogramm müssen diese Merker bearbeitet und ausgewertet werden. Der Zugriff erfolgt über deren symbolischen Namen.
- Struktur der Ein- und Ausgänge
  Diese können im Fenster [Steuerungskonfiguration] (→ Bild unten) direkt symbolisch bezeichnet werden (sehr empfohlen!) und stehen als globale Variablen im gesamten Projekt zur Verfügung.



© ifm

CAN-Schnittstellen deklarieren (z.B. CR1080)

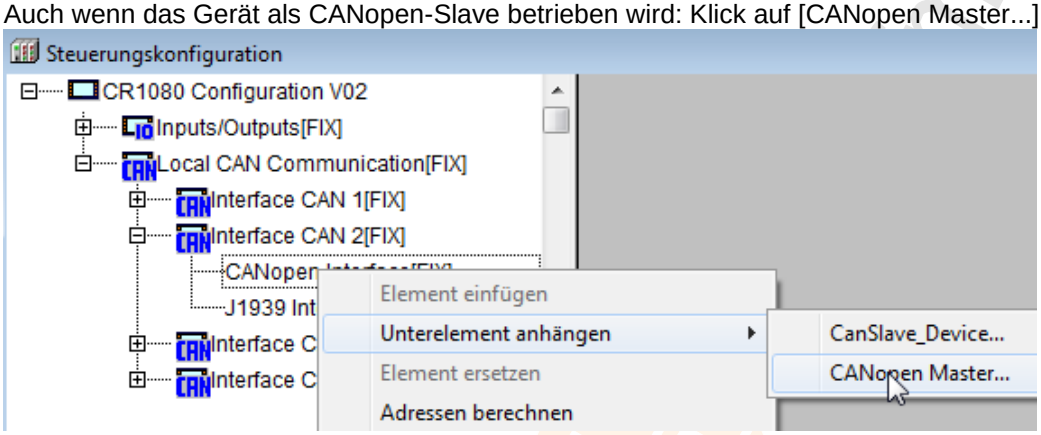
10080

In der CODESYS-Steuerungskonfiguration müssen Sie nun die CAN-Schnittstelle(n) deklarieren.

► Rechtsklick auf [CANopen Interface [FIX]] der gewünschten CAN-Schnittstelle.

► Klick auf [Unterelement anhängen].

► Auch wenn das Gerät als CANopen-Slave betrieben wird: Klick auf [CANopen Master...]:

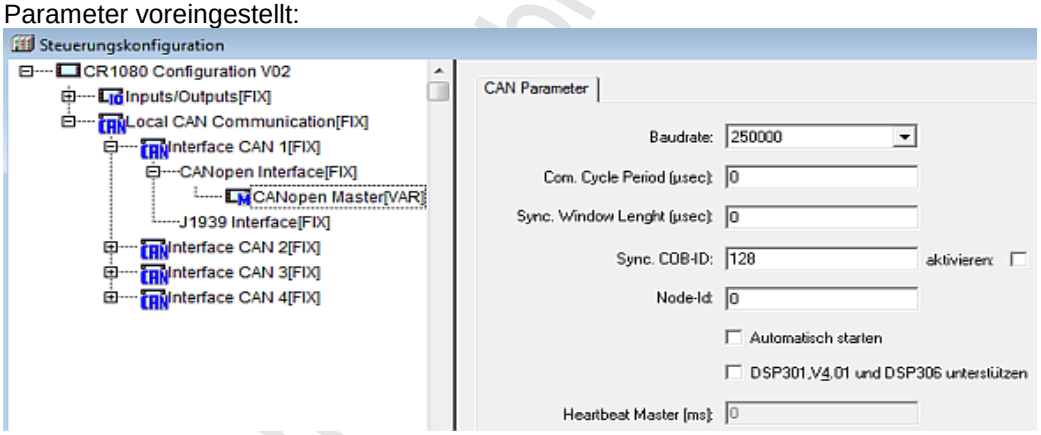


Info

Wenn das Gerät als Slave betrieben wird, wäre die Auswahl [CanSlave_Device] ebenfalls möglich.

Bei der insgesamt einfacheren Konfiguration als Master können auch alle CAN-Layer2- und Netzwerkvariablen-Funktionen genutzt werden.

> Die CAN-Parameter der Steuerungskonfiguration erscheinen. Hier sind bereits einige CAN-Parameter voreingestellt:



► Wenn das Gerät via Netzwerkvariablen oder CAN_RX / CAN_TX auf CAN-Layer 2 oder als Slave betrieben wird:
Prüfen, ob für das Gerät die richtige Baudrate eingestellt ist (Baudrate muss für alle Teilnehmer identisch sein).

► Wenn das Gerät als CANopen-Master betrieben wird:
Alle Parameter-Einstellungen prüfen.

► Das Fenster [Steuerungskonfiguration] schließen.

► Mit Menü [Datei] > [speichern unter...] dem Projekt einen sinnvollen Namen geben und das Projekt im gewünschten Verzeichnis speichern.

► Im Anwendungsprogramm für jede CAN-Schnittstelle immer eine eigene Instanz des FBs ***CANOPEN_ENABLE*** (→ Seite 94) anlegen!

4.2.2 Programmiersystem über Templates einrichten

13745

ifm bietet vorgefertigte Templates (Programm-Vorlagen), womit Sie das Programmiersystem schnell, einfach und vollständig einrichten können.

970

Beim Installieren der ecomatmobile-DVD "Software, tools and documentation" wurden auch Projekte mit Vorlagen auf Ihrem Computer im Programmverzeichnis abgelegt:
`…\ifm electronic\CoDeSys V…\Projects\Template_DVD_V…`

► Die gewünschte dort gespeicherte Vorlage in CODESYS öffnen mit:
[Datei] > [Neu aus Vorlage…]

> CODESYS legt ein neues Projekt an, dem der prinzipielle Programmaufbau entnommen werden kann. Es wird dringend empfohlen, dem gezeigten Schema zu folgen.

4.3 Funktionskonfiguration, allgemein

Inhalt

Systemvariablen 46

3971

4.3.1 Systemvariablen

15576

Alle Systemvariablen (→ Kapitel *Systemmerker* (→ Seite 214)) liegen auf festen, nicht verschiebbaren Adressen.

© ifm electronic gmbh www.ifm.com

4.4 Funktionskonfiguration der Ein- und Ausgänge

Inhalt

Konfiguration der Ein- und Ausgänge (Voreinstellung) 47
Eingänge konfigurieren 48
Ausgänge konfigurieren 52

7995
1394

Bei bestimmten Ein- und Ausgängen sind zusätzliche Diagnosefunktionen aktivierbar. Damit kann das jeweilige Ein- und Ausgangssignal überwacht werden und im Fehlerfall kann das Anwendungsprogramm darauf reagieren.

Je nach Ein- und Ausgang müssen bei der Nutzung der Diagnose bestimmte Randbedingungen beachtet werden:

► Anhand des Datenblattes prüfen, für welche Ein- und Ausgänge des Geräts welche Diagnosemöglichkeit zur Verfügung steht!

- Zur Konfiguration der Ein- und Ausgänge sind in den Gerätebibliotheken (`ifm_CR0411_Vxxyyzz.LIB`) Konstanten vordefiniert (z.B. IN_DIGITAL_H). Ausführliche Angaben → Kapitel ***Mögliche Betriebsarten Ein-/Ausgänge*** (→ Seite 216).

4.4.1 Konfiguration der Ein- und Ausgänge (Voreinstellung)

2249

- Alle Ein-/Ausgänge sind im Auslieferungszustand im Binär-Modus (plus-schaltend!).
- Die Diagnosefunktion ist nicht aktiv.
- Der Überlastschutz ist aktiv.

© ifm electronic gmbh www.ifm.com

4.4.2 Eingänge konfigurieren

Inhalt

Sicherheitshinweise zu Reed-Relais 48
Software-Filter der Eingänge konfigurieren 49
Analogeingänge: Konfiguration und Diagnose 50
Binäreingänge: Konfiguration und Diagnose 50
Schnelle Eingänge 51

3973

Zulässige Betriebsarten → Kapitel ***Mögliche Betriebsarten Ein-/Ausgänge*** (→ Seite 216)

Sicherheitshinweise zu Reed-Relais

7348

Beim Einsatz von nichtelektronischen Schaltern Folgendes beachten:

Kontakte von Reed-Relais können (reversibel) verkleben, wenn sie ohne Vorwiderstand an den Geräte-Eingängen angeschlossen werden.

- **Abhilfe:** Vorwiderstand zum Reed-Relais installieren:
  Vorwiderstand = max. Eingangsspannung / zulässiger Strom im Reed-Relais
  Beispiel: 32 V / 500 mA = 64 Ohm
- Der Vorwiderstand darf 5 % des Eingangswiderstands RE des Geräte-Eingangs (→ Datenblatt) nicht überschreiten. Sonst wird das Signal nicht als TRUE erkannt.
  Beispiel:
  RE = 3 000 Ohm
  ⇨ max. Vorwiderstand = 150 Ohm

© ifm electronic gmbh

Software-Filter der Eingänge konfigurieren

15418

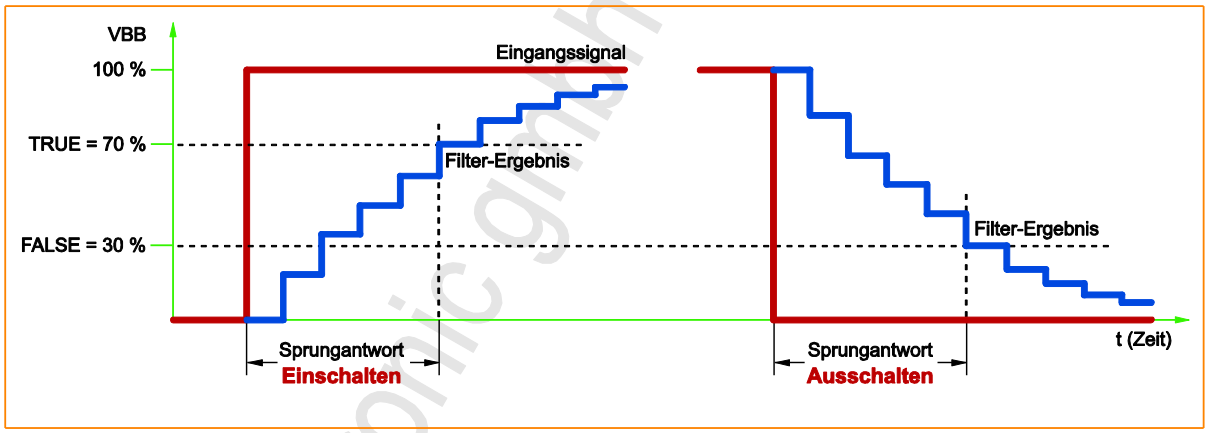
Über den Eingang FILTER im FB ***INPUT*** (→ Seite 176) kann ein Software-Filter konfiguriert werden, das die gemessene Eingangsspannung an den Analogeingängen filtert.

Das Filter verhält sich wie ein Tiefpassfilter, wobei die Filterfrequenz durch den in FILTER eingetragenen Wert eingestellt wird. Für FILTER sind Werte von 0...8 zulässig.

Tabelle: Grenzfrequenz Software-Tiefpassfilter am Analogeingang

FILTER	Filterfrequenz [Hz]	Sprungantwort [ms] für …			Hinweise
		0…70 %	0…90 %	0…99 %	
0	Filter deaktiviert				
1	120	2	4	7	
2	47	5	9	17	
3	22	10	18	35	
4	10	19	36	72	empfohlen
5	5	38	73	146	
6	2,5	77	147	293	
7	1,2	154	294	588	
8	0,7	308	589	1177	

Folgende Aussagen der Sprungantwort sind relevant:
- Input analog: 0...90 % und 0...99 %
- Input binär: 0...70 %



Grafik: Zeitverlauf binäres Signal am Eingang beim Einschalten / beim Ausschalten

© ifm electronic gmbh www.ifm.com

Analogeingänge: Konfiguration und Diagnose

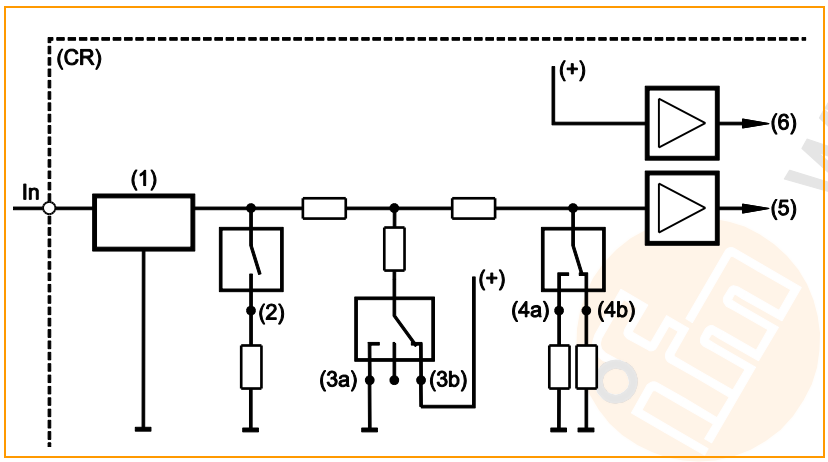
14656

Die Konfiguration jedes einzelnen Eingangs erfolgt über das Anwendungsprogramm:
• FB ***INPUT*** (→ Seite 176) > Eingang MODE

► Werden die Analogeingänge auf Strommessung konfiguriert, wird bei Überschreiten des Endwertes (23 mA für ≥ 40 ms) in den sicheren Spannungsmessbereich (0...32 V DC) geschaltet und im FB INPUT der Ausgang RESULT entsprechend gesetzt. Nach etwa einer Sekunde schaltet der Eingang selbsttätig auf den Strommessbereich zurück.

Alternativ kann ein Analog-Kanal auch binär ausgewertet werden.

8971



In = Anschluss Multifunktions-Eingang n
(CR) = Gerät
(1) = Eingangsfilter
(2) = analoge Strommessung
(3a) = Binär-Eingang plus-schaltend
(3b) = Binär-Eingang minus-schaltend
(4a) = analoge Spannungsmessung 0...10 V
(4b) = analoge Spannungsmessung 0...32 V
(5) = Spannung
(6) = Referenz-Spannung

Grafik: Prinzipschaltung Multifunktions-Eingang

Binäreingänge: Konfiguration und Diagnose

16647

Die Konfiguration jedes einzelnen Eingangs erfolgt über das Anwendungsprogramm:
• FB ***INPUT*** (→ Seite 176) > Eingang MODE

MODE	BYTE	Betriebsart des Eingangskanals:		
		0 = 0x00	Aus	
		1 = 0x01	(nur für binär ausgewertete Eingänge) Binäreingang, plus-schaltend (BL)	
		3 = 0x03	Spannungseingang	0…10 000 mV
		6 = 0x06	Spannungseingang, ratiometrisch	0...1 000 ‰
		7 = 0x07	Stromeingang	0…20 000 µA
		9 = 0x09	Spannungseingang	0…32 000 mV
		10 = 0x0A	(nur für analog ausgewertete Eingänge) Binäreingang, plus-schaltend (BL)	
		11 = 0x0B	(nur für analog ausgewertete Eingänge) Binäreingang, plus-schaltend (BL) mit Diagnose (Namur)	
		12 = 0x0C	Binäreingang, minus-schaltend (BH)	
		18 = 0x12	Widerstandseingang	16…30 000 Ω

© ifm electronic gmbh www.ifm.com

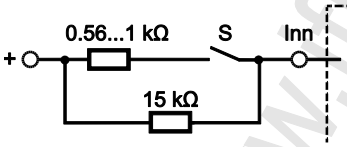
Diagnose der Eingänge aktivieren

7352

Soll die Diagnose genutzt werden, muss diese zusätzlich aktiviert werden.

► Am Eingang MODE des FB ***INPUT*** (→ Seite 176) die Betriebsart des Eingangs einstellen.

> Der FB ***INPUT*** (→ Seite 176) liefert die Diagnosemeldungen der Eingänge an seinem Ausgang RESULT.

Diagnose bei nichtelektronischen Schaltern: ► Schalter mit einer zusätzlichen Widerstandsbeschaltung versehen!	 Grafik: Nichtelektronischer Schalter S am Eingang Inn

An diesen Eingängen können diagnosefähige Sensoren nach NAMUR verwendet werden. Eine zusätzliche Widerstandsbeschaltung ist dann nicht erforderlich.

Schnelle Eingänge

8292

Die Geräte verfügen über schnelle Zähl-/Impulseingänge für eine Eingangsfrequenz bis 30 kHz (→ Datenblatt).

Werden z.B. mechanische Schalter an diesen Eingängen angeschlossen, kann es durch Kontaktprellen zu Fehlsignalen in der Steuerung kommen.

Geeignete Funktionsbausteine sind z.B.:

FAST_COUNT	Zählerbaustein für schnelle Eingangsimpulse
INC_ENCODER (→ Seite 173)	Vorwärts-/Rückwärts-Zählerfunktion zur Auswertung von Drehgebern
PERIOD (→ Seite 178)	misst am angegebenen Kanal die Frequenz und die Periodendauer (Zykluszeit) in [µs]

Bei Einsatz dieser Bausteine werden automatisch die dort parametrierten Ein-/Ausgänge konfiguriert. Der Programmierer der Anwendung ist hiervon entlastet.

Einsatz als Binäreingänge

3804

Durch die zulässigen hohen Eingangsfrequenzen können auch Fehlsignale erkannt werden, z.B. prellende Kontakte mechanischer Schalter.

► Bei Bedarf die Fehlsignale im Anwendungsprogramm unterdrücken!

© ifm electronic gmbh www.ifm.com

4.4.3 Ausgänge konfigurieren

Inhalt

Software-Filter der Ausgänge konfigurieren 52
Binärausgänge: Konfiguration und Diagnose 53
PWM-Ausgänge 54

3976

Zulässige Betriebsarten → Kapitel ***Mögliche Betriebsarten Ein-/Ausgänge*** (→ Seite 216)

Software-Filter der Ausgänge konfigurieren

15421

Über den Eingang FILTER im FB ***OUTPUT*** (→ Seite 183) kann ein Software-Filter konfiguriert werden, das den gemessene Ausgangsstrom an den PWM-Ausgängen filtert.

Das FILTER-Byte gilt nur für Ausgänge mit Strommessung.
Für Ausgänge ohne Strommessung: FILTER = 0 setzen!

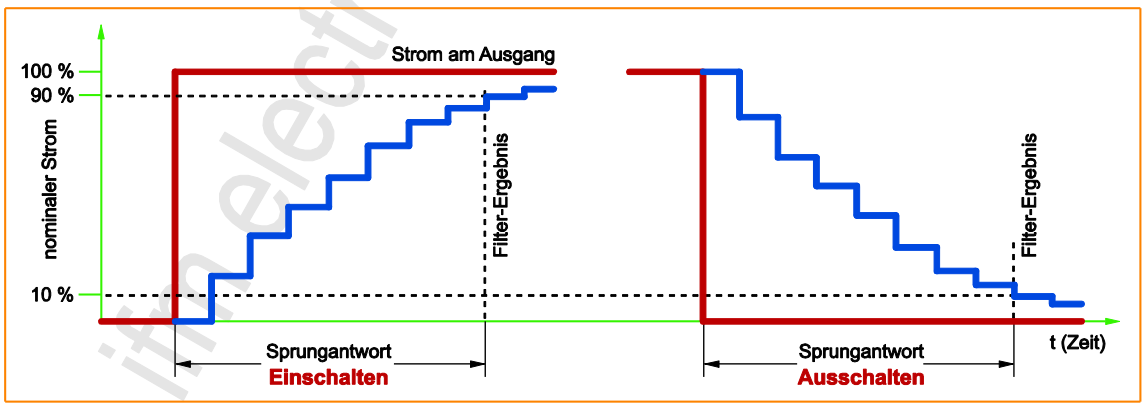
Der Strom am Ausgang wird über eine PWM-Periode gemittelt.
Falls Dithering eingestellt ist, wird der Strom über die Ditherperiode gemittelt.

Das Filter verhält sich wie ein Tiefpassfilter, wobei die Grenzfrequenz durch den in FILTER eingetragenen Wert eingestellt wird. Für FILTER sind Werte von 0...8 zulässig.

Tabelle: Grenzfrequenz Software-Tiefpassfilter am PWM-Ausgang

FILTER	Filterfrequenz [Hz]	Sprungantwort [ms] für …			Hinweise
		0…90 %	0…95 %	0…99 %	
0	Filter deaktiviert				Ausgänge ohne Strommessung
1	600	0,8	1,0	1,4	
2	233	1,8	2,2	3,4	
3	109	3,6	4,6	7,0	
4	52	7,2	9,4	14,4	empfohlen
5	26	14,6	19,0	29,2	
6	13	29,4	38,2	58,6	
7	6	58,8	76,4	117,6	
8	4	117,8	153,2	235,4	

Folgende Aussagen der Sprungantwort sind relevant:
• Output current: 0...90 % und 0...99 %



Grafik: Zeitverlauf binäres Stromsignal am Ausgang beim Einschalten / beim Ausschalten

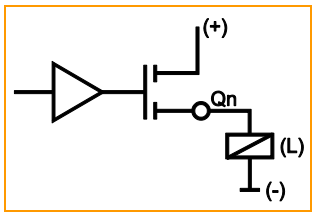
© ifm electronic gmbh www.ifm.com

Binärausgänge: Konfiguration und Diagnose

15452

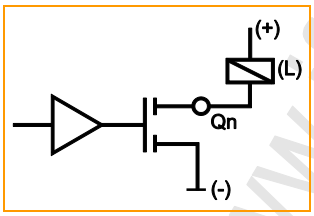
Bei den Geräte-Ausgängen sind folgende Betriebsarten möglich (→ Datenblatt):
- binärer Ausgang, plus-schaltend (BH) mit/ohne Diagnosefunktion
- binärer Ausgang, minus-schaltend (BL) ohne Diagnosefunktion

15450



Qn = Anschluss Ausgang n
(L) = Last

Prinzipschaltung Binär-Ausgang plus-schaltend (BH)
für positives Ausgangssignal



Qn = Anschluss Ausgang n
(L) = Last

Prinzipschaltung Binär-Ausgang minus-schaltend (BL)
für negatives Ausgangssignal

► Die Konfiguration jedes einzelnen Ausgangs erfolgt über das Anwendungsprogramm:
→ FB ***OUTPUT*** (→ Seite 183) > Eingang MODE.

⚠ WARNUNG

Gefährlicher Wiederanlauf möglich!
Gefahr von Personenschaden! Gefahr von Sachschaden an der Maschine/Anlage!

Wird ein Ausgang im Fehlerfall hardwaremäßig abgeschaltet, ändert sich der durch das Anwendungsprogramm erzeugte logische Zustand dadurch nicht.

► Abhilfe:
- Die Ausgänge zunächst im Anwendungsprogramm logisch zurücksetzen!
- Fehler beseitigen!
- Ausgänge situationsabhängig wieder setzen.

Diagnose der Ausgänge konfigurieren

8301

Soll die Diagnose genutzt werden, muss diese zusätzlich aktiviert werden.

► Ausgang als Binärausgang mit Diagnose nutzen (→ Datenblatt):
→ FB OUTPUT > Eingang MODE = 15 oder 16

> Der FB ***OUTPUT*** (→ Seite 183) liefert die Diagnosemeldungen der Ausgänge an seinem Ausgang RESULT.

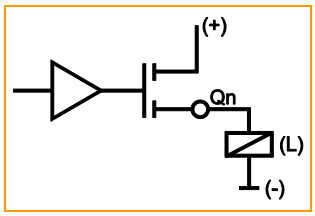
© ifm electronic gmbh www.ifm.com

PWM-Ausgänge

16284

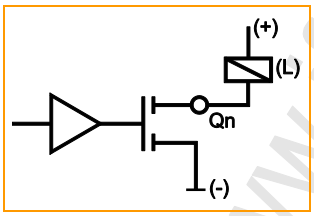
Bei den Geräte-Ausgängen sind folgende Betriebsarten möglich (→ Datenblatt):
- PWM-Ausgang, plus-schaltend (BH) ohne Diagnosefunktion
- PWM-Ausgangspaar H-Brücke ohne Diagnosefunktion

15450



Qn = Anschluss Ausgang n
(L) = Last

Prinzipschaltung Binär-Ausgang plus-schaltend (BH)
für positives Ausgangssignal



Qn = Anschluss Ausgang n
(L) = Last

Prinzipschaltung Binär-Ausgang minus-schaltend (BL)
für negatives Ausgangssignal

16253

⚠ WARNUNG

Sach- oder Körperschäden möglich durch Fehlfunktionen!

Für Ausgänge im PWM-Modus gilt:
- es gibt keine Diagnosefunktionen

9980

HINWEIS

PWM-Ausgänge dürfen NICHT parallel betrieben werden, um z.B. den max. Ausgangsstrom zu erhöhen. Die Ausgänge arbeiten nicht synchron.

Andernfalls kann die komplette Last über nur einen Ausgang gehen. Die Strommessung funktioniert dann nicht mehr.

- PWM-Ausgänge können mit und ohne Stromregelfunktion betrieben werden.
  Stromgeregelte PWM-Ausgänge werden überwiegend zur Ansteuerung von proportionalen Hydraulikfunktionen genutzt.

Verfügbarkeit von PWM

16285

Gerät	Anzahl verfügbare PWM-Ausgänge	davon stromgeregelt (PWMi)	PWM-Frequenz [Hz]
BasicController: CR0411	8	8	20...250

© ifm electronic gmbh www.ifm.com

FBs für PWM-Funktionen

16387

Für die PWM-Funktion der Ausgänge stehen folgende Funktionsbausteine zur Verfügung:

CURRENT_CONTROL (→ Seite 181)	Stromregler für einen PWMi-Ausgangskanal
H_BRIDGE (→ Seite 189)	H-Brücke an einem PWM-Kanalpaar
PWM1000 (→ Seite 185)	initialisiert und parametriert einen PWM-fähigen Ausgangskanal das Puls-Pausen-Verhältnis kann in 1 ‰-Schritten angegeben werden
PWM1000_LOW (→ Seite 187)	initialisiert und parametriert einen PWM-fähigen Ausgangskanal minus-schaltend das Puls-Pausen-Verhältnis kann in 1 ‰-Schritten angegeben werden

Stromregelung mit PWM (= PWMi)

14722

Über die im Controller integrierten Strommesskanäle kann eine Strommessung des Spulenstroms durchgeführt werden. Dadurch kann zum Beispiel der Strom bei einer Spulenerwärmung nachgeregelt werden. Damit bleiben die Hydraulikverhältnisse im System gleich.

Grundsätzlich sind die stromgeregelten Ausgänge gegen Kurzschluss geschützt.

© ifm electronic gmbh www.ifm.com

4.5 Variablen

Inhalt

Retain-Variablen 57
Netzwerkvariablen 58

3130

In diesem Kapitel erfahren Sie mehr über den Umgang mit Variablen.
Grundsätzlich unterscheiden wir in CODESYS folgende Arten von Variablen:

3130

Das Gerät unterstützt folgende Variablentypen:

Variable	Deklarationsort	Gültigkeitsbereich	Speicherverhalten
lokal	im Deklarationsteil des Bausteins	gilt nur im Baustein (POU), in dem sie konfiguriert wurde	flüchtig
lokal Retain			nicht flüchtig
global	in [Ressourcen] > [Globale Variablen] > [Globale_Variablen]	gilt in allen Bausteinen (POUs) dieses CODESYS-Projekts	flüchtig
global Retain			nicht flüchtig
Netzwerk	in [Ressourcen] > [Globale Variablen] > Deklarationsliste	Werte stehen allen CODESYS-Projekten im gesamten Netzwerk zur Verfügung, wenn die Variable in ihren Deklarationslisten enthalten ist.	flüchtig
Netzwerk Retain			nicht flüchtig

→ CODESYS-Programmierhandbuch → ecomat*mobile*-DVD "Software, tools and documentation"

© ifm electronic gmbh www.ifm.com

4.5.1 Retain-Variablen

8672

Retain-Variablen werden automatisch in einen geschützten Speicherbereich gesichert und automatisch bei Neustart wieder geladen.

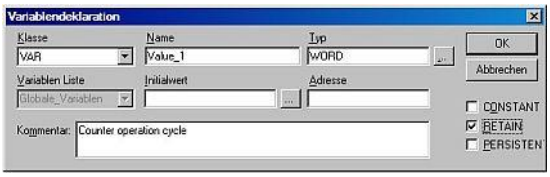
14166

Typische Einsätze für Retain-Variablen sind z.B.:
- Betriebsstunden, die zur Laufzeit der Maschine fortgeschrieben werden,
- Positionswerte von Inkrementalgebern,
- im Bildschirmgerät eingetragene Sollwerte,
- Maschinenparameter,

also alle Variablen, deren Werte beim Ausschalten des Geräts nicht verloren gehen dürfen.

Als Retain können alle Variablentypen, auch komplexe Stukturen (z.B. Timer), gekennzeichnet werden.

► Dazu in der Variablen-Deklaration das Kontrollfeld [RETAIN] aktivieren (→ Bild).



Sichern von Retain-Variablen

9853

Im Gerät werden auch die Daten vom Typ RETAIN zur Laufzeit nur im flüchtigen Speicher (RAM) abgelegt.
Um sie dauerhaft zu sichern, werden sie am Ende jedes Zyklus automatisch in den FRAM-Speicher [1]) geschrieben.

[1]) FRAM steht hier allgemein für alle Arten von nichtflüchtigen, schnellen Speichern.

HINWEIS

In diesem Gerät die folgenden Funktionen aus der 3S-Bibliothek `SysLibPlcCtrl.lib` NICHT einsetzen:
- FUN SysSaveRetains
- FUN SysRestoreRetains

Rücklesen von Retain-Variablen

9854

Nach dem Einschalten und vor dem ersten Programmzyklus schreibt das Gerät die gesicherten Daten einmalig automatisch zurück in den Arbeitsspeicher. Dazu müssen keine zusätzlichen Bausteine in das Anwendungsprogramm integriert werden.

HINWEIS

In diesem Gerät die folgenden Funktionen aus der 3S-Bibliothek `SysLibPlcCtrl.lib` NICHT einsetzen:
- FUN SysSaveRetains
- FUN SysRestoreRetains

© ifm electronic gmbh www.ifm.com

4.5.2 Netzwerkvariablen

15242
9856

Globale Netzwerkvariablen dienen dem Datenaustausch zwischen Controllern im Netzwerk. Die Werte von globalen Netzwerkvariablen stehen allen CODESYS-Projekten im gesamten Netzwerk zur Verfügung, wenn die Variablen in deren Deklarationslisten enthalten sind.

► Dazu folgende Bibliothek(en) in das CODESYS-Projekt einbinden:
- `3S_CANopenNetVar.lib`
- `ifm_NetVarLib_NT_Vxxyyzz.lib`

© ifm electronic gmbh www.ifm.com

5 ifm-Funktionselemente

Inhalt

ifm-Bibliotheken für das Gerät CR0411 59
ifm-Bausteine für das Gerät CR0411 65

13586

Alle CODESYS-Funktionselemente (FBs, PRGs, FUNs) sind in Bibliotheken zusammengefasst. Nachfolgend zeigen wir Ihnen alle ifm-Bibliotheken, die Sie zusammen mit diesem Gerät nutzen können.

Anschließend finden Sie eine thematisch gegliederte Beschreibung der Funktionselemente.

5.1 ifm-Bibliotheken für das Gerät CR0411

Inhalt

Bibliothek ifm_CR0411_V03yyzz.LIB 60
Bibliothek ifm_RAWCan_NT_Vxxyyzz.LIB 61
Bibliothek ifm_CANopen_NT_Vxxyyzz.LIB 62
Bibliothek ifm_J1939_NT_Vxxyyzz.LIB 63
Bibliothek ifm_CR0411_util_Vxxyyzz.LIB 64

14235

Legende für ..._Vxxyyzz.LIB:

V	Version
xx: 00...99	Versionsnummer
yy: 00...99	Release-Nummer
zz: 00...99	Patch-Nummer

Hier finden Sie die für dieses Gerät passenden ifm-Funktionselemente aufgelistet, nach CODESYS-Bibliotheken sortiert.

© ifm electronic gmbh

5.1.1 Bibliothek ifm_CR0411_V03yyzz.LIB

16397

Dies ist die Geräte-Bibliothek. Diese ifm-Bibliothek enthält folgende Bausteine:

Baustein	Kurzbeschreibung
CURRENT_CONTROL (→ Seite 181)	Stromregler für einen PWMi-Ausgangskanal
FASTCOUNT (→ Seite 171)	Zählerbaustein für schnelle Eingangsimpulse
FLASH_INFO (→ Seite 195)	liest die Informationen aus dem User-Flash-Speicher: • Name des Speicherbereichs (vom User vorgegeben), • Software-Version, • Startadresse (für einfaches Lesen mit IEC-Struktur)
FLASH_READ (→ Seite 196)	liest unterschiedliche Datentypen direkt aus dem Flash-Speicher in den RAM
GET_APP_INFO (→ Seite 197)	liefert Informationen über das im Gerät gespeicherte Anwendungsprogramm: • Name der Anwendung, • Version der Anwendung, • eindeutige CODESYS-Build-Nummer, • CODESYS-Build-Datum
GET_HW_INFO (→ Seite 198)	liefert Informationen über die Hardware des Geräts: • ifm-Artikelnummer (z.B. CR0403), • Artikelbezeichnung, • eindeutige Seriennummer, • Hardware-Revision, • Produktionsdatum
GET_IDENTITY (→ Seite 199)	liest die im Gerät gespeicherte Kennung der Anwendung (wurde zuvor mit ***SET_IDENTITY*** (→ Seite 207) gespeichert)
GET_SW_INFO (→ Seite 200)	liefert Informationen über die System-Software des Geräts: • Software-Name, • Software-Version, • Build-Nummer, • Build-Datum
GET_SW_VERSION (→ Seite 201)	liefert Informationen über die Versionen der im Gerät gespeicherten Software: • Basic-System-Version, • Bootloader-Version, • SIS-Version, • Anwendungsprogramm-Version, • User-Flash-Version
INC_ENCODER (→ Seite 173)	Vorwärts-/Rückwärts-Zählerfunktion zur Auswertung von Drehgebern
INPUT (→ Seite 176)	weist einem Eingangskanal eine Betriebsart zu liefert den aktuellen Zustand am gewählten Kanal
MEM_ERROR (→ Seite 202)	meldet Fehler in einigen Parametern oder im Speicher (Re-)Initialisierung von Systemressourcen
MEMCPY (→ Seite 203)	schreibt und liest unterschiedliche Datentypen direkt in den Speicher
OHC (→ Seite 205)	parametrierbarer Betriebsstunden-Zähler (0...3)
OUTPUT (→ Seite 183)	weist einem Ausgangskanal eine Betriebsart zu liefert den aktuellen Zustand am gewählten Kanal
PERIOD (→ Seite 178)	misst am angegebenen Kanal die Frequenz und die Periodendauer (Zykluszeit) in [µs]
PWM1000 (→ Seite 185)	initialisiert und parametriert einen PWM-fähigen Ausgangskanal das Puls-Pausen-Verhältnis kann in 1 ‰-Schritten angegeben werden
PWM1000_LOW (→ Seite 187)	initialisiert und parametriert einen PWM-fähigen Ausgangskanal minus-schaltend das Puls-Pausen-Verhältnis kann in 1 ‰-Schritten angegeben werden
SET_IDENTITY (→ Seite 207)	setzt eine anwendungsspezifische Programmkennung
SET_LED (→ Seite 208)	im Anwendungsprogramm Frequenz und Farbe der Status-LED ändern
SET_PASSWORD (→ Seite 210)	setzt Benutzerkennung für Zugangskontrolle bei Programm- und Speicher-Upload
TIMER_READ_US (→ Seite 211)	liest die aktuelle Systemzeit in [µs] aus Max-Wert = 1h 11min 34s 967ms 295µs

© ifm electronic gmbH www.ifm.com

5.1.2 Bibliothek ifm_RAWCan_NT_Vxxyyzz.LIB

14715

Diese ifm-Bibliothek enthält folgende Bausteine:

Baustein	Kurzbeschreibung
CAN_ENABLE (→ Seite 68)	initialisiert die angegebene CAN-Schnittstelle parametriert die CAN-Baudrate
CAN_RECOVER (→ Seite 69)	das automatische BusOff-Handling aktivieren / deaktivieren bei Vorliegen eines BusOff die CAN-Schnittstelle wieder neu starten
CAN_REMOTE_REQUEST (→ Seite 90)	eine entsprechende Anforderung senden und die Antwort des anderen Gerätes als Ergebnis zurückliefern
CAN_REMOTE_RESPONSE (→ Seite 91)	stellt dem CAN-Controller im Gerät Daten zur Verfügung, die automatisch auf die Anfrage einer Remote-Nachricht gesendet werden
CAN_RX (→ Seite 74)	konfiguriert ein Datenempfangsobjekt und liest den Empfangspuffer des Datenobjektes aus
CAN_RX_ENH (→ Seite 75)	• konfiguriert ein Datenempfangsobjekt und liest den Empfangspuffer des Datenobjektes aus • Frame-Typ und Maske wählbar
CAN_RX_ENH_FIFO (→ Seite 77)	• konfiguriert ein Datenempfangsobjekt und liest den Empfangspuffer des Datenobjektes aus • Frame-Typ und Maske wählbar • mehrere CAN-Telegramme je Zyklus möglich
CAN_RX_RANGE (→ Seite 79)	• konfiguriert einen Bereich von Datenempfangsobjekten und liest den Empfangspuffer der Datenobjekte aus • Frame-Typ und Maske wählbar
CAN_RX_RANGE_FIFO (→ Seite 81)	• konfiguriert einen Bereich von Datenempfangsobjekten und liest den Empfangspuffer der Datenobjekte aus • Frame-Typ und Maske wählbar • mehrere CAN-Telegramme je Zyklus möglich
CAN_SETDOWNLOADID (→ Seite 70)	= Set CAN-Download-ID stellt den Download-Identifier für die CAN-Schnittstelle ein
CAN_STATUS (→ Seite 71)	Informationen zum gewählten CAN-Bus abfragen: BAUDRATE, DOWNLOAD_ID, BUSOFF, WARNING_RX, WARNING_TX, VERSION, BUSLOAD und bei Bedarf zurücksetzen: BUSOFF, WARNING_RX, WARNING_TX
CAN_TX (→ Seite 84)	übergibt in jedem Aufruf ein CAN-Datenobjekt (Message) an die parametrierte CAN-Schnittstelle zur Übertragung
CAN_TX_ENH (→ Seite 85)	übergibt in jedem Aufruf ein CAN-Datenobjekt (Message) an die parametrierte CAN-Schnittstelle zur Übertragung CAN-spezifische Eigenschaften sind einstellbar
CAN_TX_ENH_CYCLIC (→ Seite 87)	übergibt zyklisch ein CAN-Datenobjekt (Message) an die parametrierte CAN-Schnittstelle zur Übertragung CAN-spezifische Eigenschaften sind einstellbar

© ifm electronic gmbh www.ifm.com

5.1.3 Bibliothek ifm_CANopen_NT_Vxxyyzz.LIB

14914

Diese ifm-Bibliothek enthält folgende Bausteine:

Baustein	Kurzbeschreibung
CANOPEN_ENABLE (→ Seite 94)	initialisiert die angegebene CANopen-Master-Schnittstelle parametriert die CAN-Baudrate
CANOPEN_GETBUFFERFLAGS (→ Seite 96)	= CANopen Get Bufferflags liefert Informationen zu den Buffer-Flags Über optionale Eingänge können die Flags zurückgesetzt werden.
CANOPEN_GETEMCYMESSAGES (→ Seite 133)	= Get CANopen Emergency Messages listet alle Emergency-Nachrichten auf, die die Steuerung seit dem letzten Löschen der Nachrichten von anderen Knoten am Netz empfangen hat Die Liste kann durch Setzen des entsprechenden Eingangs zurückgesetzt werden.
CANOPEN_GETERRORREGISTER (→ Seite 135)	= Get CANopen Error-Register liest die Fehler-Register 0x1001 und 0x1003 der Steuerung aus Die Register können durch Setzen der entsprechenden Eingänge zurückgesetzt werden.
CANOPEN_GETGUARDHBERRLIST (→ Seite 129)	= Get CANopen-Guard and Heartbeat Error-List listet in einem Array alle Knoten auf, für die der Master einen Fehler erkannt hat: Guarding-Fehler, Heartbeat-Fehler Die Liste kann durch Setzen des entsprechenden Eingangs zurückgesetzt werden.
CANOPEN_GETGUARDHBSTATSLV (→ Seite 130)	= CANopen-Slave get Guard and Heartbeat State meldet der Steuerung im Slave-Betrieb folgende Zustände: Node-Guarding-Überwachung, Heartbeat-Überwachung Die gemeldeten Fehler können durch Setzen des entsprechenden Eingangs zurückgesetzt werden.
CANOPEN_GETNMTSTATESLAVE (→ Seite 103)	= CANopen-Slave get Network Management State meldet den Netzwerk-Betriebszustand des Knotens
CANOPEN_GETODCHANGEDFLAG (→ Seite 107)	= Get Object Directory Changed Flag meldet wenn bei einem bestimmten Objektverzeichnis-Eintrag der Wert geändert wurde
CANOPEN_GETSTATE (→ Seite 98)	= CANopen Set State Parameter des Masters, eines Slave-Devices oder eines bestimmten Knotens im Netz abfragen
CANOPEN_GETSYNCSTATE (→ Seite 125)	= CANopen get SYNC State • liest die Einstellung der SYNC-Funktionalität (aktiv / deaktiv), • liest den Fehlerzustand der SYNC-Funktionalität (SyncError)
CANOPEN_NMTSERVICES (→ Seite 104)	= CANopen Network Management Services aktualisiert den internen Knotenstatus sowie abhängig von den NMT-Kommando-Eingängen: • löst ein NMT-Kommando aus oder • löst die Initialisierung eines Knotens aus
CANOPEN_READOBJECTDICT (→ Seite 108)	= CANopen Read Object Directory liest Konfigurationsdaten aus dem Objektverzeichnis des Geräts
CANOPEN_SDOREAD (→ Seite 112)	= CANopen Read SDO liest ein "expedited SDO" = beschleunigtes Nachrichten-Objekt mit Servicedaten
CANOPEN_SDOREADBLOCK (→ Seite 114)	= CANopen Read SDO Block liest den angegeben Eintrag im Objektverzeichnis eines Knotens im Netz per SDO-Blocktransfer
CANOPEN_SDOREADMULTI (→ Seite 116)	= CANopen Read SDO Multi liest den angegeben Eintrag im Objektverzeichnis eines Knotens im Netz
CANOPEN_SDOWRITE (→ Seite 118)	= SDO Write schreibt ein "expedited SDO" = beschleunigtes Nachrichten-Objekt mit Servicedaten
CANOPEN_SDOWRITEBLOCK (→ Seite 120)	= CANopen Write SDO Block schreibt in den angegeben Eintrag im Objektverzeichnis eines Knotens im Netz per SDO-Blocktransfer
CANOPEN_SDOWRITEMULTI (→ Seite 122)	= CANopen Write SDO Multi schreibt in den angegeben Eintrag im Objektverzeichnis eines Knotens im Netz

© ifm electronic gmbh www.ifm.com

Baustein	Kurzbeschreibung
CANOPEN_SENDEMCYMESSAGE (→ Seite 136)	= CANopen Send Emergency-Message versendet eine EMCY-Nachricht. Die Nachricht wird aus den entsprechenden Parametern zusammengebaut und ins Register 0x1003 eingetragen
CANOPEN_SETSTATE (→ Seite 100)	= CANopen Set State Parameter des Masters, eines Slave-Devices oder eines bestimmten Knotens im Netz setzen
CANOPEN_SETSYNCSTATE (→ Seite 127)	= CANopen Set SYNC State die SYNC-Funktionalität ein- und ausschalten
CANOPEN_WRITEOBJECTDICT (→ Seite 109)	= CANopen Write Object Directory schreibt Konfigurationsdaten in das Objektverzeichnis des Geräts

5.1.4 Bibliothek ifm_J1939_NT_Vxxyyzz.LIB

14912

Diese ifm-Bibliothek enthält folgende Bausteine:

Baustein	Kurzbeschreibung
J1939_DM1RX (→ Seite 163)	J1939 Diagnostic Message 1 RX empfängt Diagnosemeldungen DM1 oder DM2 von anderen ECUs
J1939_DM1TX (→ Seite 165)	J1939 Diagnostic Message 1 TX eine aktive Fehlermeldung an den CAN-Stack übergeben
J1939_DM1TX_CFG (→ Seite 168)	J1939 Diagnostic Message 1 TX configurable CAN-Stack sendet keine zyklischen DM1 "zero active faults"-Nachrichten
J1939_DM3TX (→ Seite 169)	J1939 Diagnostic Message 3 TX löscht inaktive DTCs (DM2) auf einem Gerät
J1939_ENABLE (→ Seite 139)	Initialisiert den J1939-Stack
J1939_GETDABYNAME (→ Seite 141)	= Get Destination Arbitrary Name anhand der Namensinformation die Ziel-Adresse eines oder mehrerer anderer Teilnehmer bestimmen
J1939_NAME (→ Seite 143)	dem Gerät einen Name geben, mit dem es sich im Netzwerk identifiziert
J1939_RX (→ Seite 150)	empfängt eine Single-Frame-Nachricht zeigt die zuletzt auf dem CAN-Bus gelesene Nachricht
J1939_RX_FIFO (→ Seite 151)	= J1939 RX with FIFO empfängt alle spezifizierten Nachrichten und liest sie nacheinander aus einem FIFO
J1939_RX_MULTI (→ Seite 153)	= J1939 RX Multiframe Message empfängt Multiframe-Nachrichten
J1939_SPEC_REQ (→ Seite 147)	= J1939 Specific Request fragt eine spezifizierte Nachricht bei einer anderen Steuerung an und empfängt sie
J1939_SPEC_REQ_MULTI (→ Seite 148)	= J1939 Specific Request Multiframe Message fragt eine spezifizierte Multiframe-Nachricht bei einer anderen Steuerung an und empfängt sie
J1939_STATUS (→ Seite 145)	zeigt relevante Informationen zum J1939-Stack
J1939_TX (→ Seite 155)	versendet einzelne Single-Frame-Nachrichten
J1939_TX_ENH (→ Seite 156)	= J1939 TX enhanced versendet einzelne Single-Frame-Nachrichten zusätzlich einstellbar: Sende-Priorität, Datenlänge
J1939_TX_ENH_CYCLIC (→ Seite 158)	= J1939 TX enhanced cyclic versendet zyklisch Single-Frame-Nachrichten zusätzlich einstellbar: Sende-Priorität, Datenlänge, Periode
J1939_TX_ENH_MULTI (→ Seite 160)	= J1939 TX enhanced Multiframe Message versendet einzelne Multiframe-Nachrichten

© ifm electronic gmbh www.ifm.com

5.1.5 Bibliothek ifm_CR0411_util_Vxxyyzz.LIB

16452

Diese ifm-Bibliothek enthält folgende Bausteine:

Baustein	Kurzbeschreibung
H_BRIDGE (→ Seite 189)	H-Brücke an einem PWM-Kanalpaar

© ifm electronic gmbh www.ifm.com

5.2 ifm-Bausteine für das Gerät CR0411

Inhalt

Baustein-Ausgänge 66
Bausteine: RAW-CAN (Layer 2) 67
Bausteine: CANopen 93
Bausteine: SAE J1939 138
Bausteine: Eingangswerte verarbeiten 170
Bausteine: Ausgangsfunktionen 180
Bausteine: System 194

13988
3826

Hier finden Sie die Beschreibung der für dieses Gerät passenden ifm-Funktionselemente, nach Thema sortiert.

© ifm electronic gmbh

5.2.1 Baustein-Ausgänge

8354
7556

Einige Bausteine geben eine Ergebnis-Meldung RESULT zurück.

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1...31		Globale Rückgabewerte; Beispiele:
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
4	04	FB ist in der Bearbeitung – Daten werden zyklisch bearbeitet
5	05	FB ist in der Bearbeitung – Empfangen läuft noch
6	06	FB ist in der Bearbeitung – Senden läuft noch
7	07	FB ist in der Bearbeitung – Remote für ID aktiv
8	08	Funktionsbaustein ist aktiv
14	0E	FB ist aktiv CANopen-Manager konfiguriert Devices und sendet SDOs
15	0F	FB ist aktiv CANopen-Manager ist gestartet
32...63		FB-spezifische Rückgabewerte
64…127		FB-spezifische Fehlermeldungen
128...255		Globale Fehlermeldungen; Beispiele:
238	EE	Fehler: CANopen-Konfiguration ist zu groß und kann nicht gestartet werden
239	EF	Fehler: CANopen-Manager konnte nicht gestartet werden
240	F0	Fehler: Mehrere modale Eingänge sind aktiv z.B. CANopen-NMT-Services
241	F1	Fehler: CANopen-Zustandsübergang ist nicht erlaubt
242	F2	Fehler: Einstellung ist nicht möglich
247	F7	Fehler: Speicherüberschreitung (Länge größer als Array)
250	FA	Fehler: FiFo ist voll – Daten wurden verloren
252	FC	Fehler: CAN-Multiframe-Sendung ist fehlgeschlagen
253	FD	Fehler: CAN-Sendung ist fehlgeschlagen. Daten können nicht gesendet werden.
255	FF	Fehler: zu wenig Speicher für Empfangs-Multiframe verfügbar

© ifm electronic gmbh www.ifm.com

5.2.2 Bausteine: RAW-CAN (Layer 2)

Inhalt

Bausteine: RAW-CAN Status 67
Bausteine: RAW-CAN Daten empfangen 73
Bausteine: RAW-CAN Daten senden 83
Bausteine: RAW-CAN Remote 89

15051

Hier beschreiben wir die RAW-CAN-Funktionsbausteine (CAN Layer 2) der **ifm electronic** zur Nutzung im Anwendungsprogramm.

Bausteine: RAW-CAN Status

Inhalt

CAN_ENABLE 68
CAN_RECOVER 69
CAN_SETDOWNLOADID 70
CAN_STATUS 71

15049

© ifm electronic gmbh

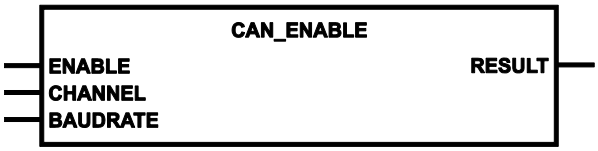
CAN_ENABLE

7492

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7494

Mit CAN_ENABLE wird die CAN-Hardware initialisiert. Ohne diesen Aufruf sind keine anderen Aufrufe im RAW-CAN möglich oder liefern einen Fehler zurück.

Zum Ändern der Baudrate ist folgender Ablauf erforderlich:

► Funktionsbaustein einen Zyklus lang auf ENABLE=FALSE halten.

> Alle Protokolle werden zurückgesetzt.

> Re-Initialisierung der CAN-Schnittstelle und auch der darauf laufenden CAN-Protokolle. Alle zum zyklischen Senden vorhanden Informationen gehen dabei ebenfalls verloren und müssen neu aufgesetzt werden.

> Beim erneuten ENABLE=TRUE wird die neue Baudrate übernommen.

Parameter der Eingänge

7495

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: CAN-Schnittstelle freigeben FALSE: CAN-Schnittstelle sperren
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
BAUDRATE	WORD := 250	Baudrate [kBit/s] zulässig = 20, 50, 100, 125, 250, 500, 800, 1000

Parameter der Ausgänge

8530

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist aktiv
9	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

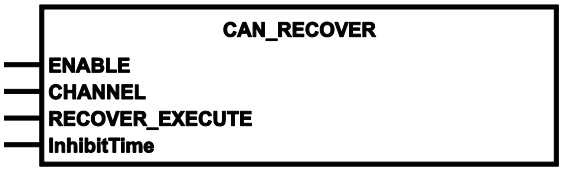
© ifm electronic gmbh www.ifm.com

CAN_RECOVER

7512

Baustein-Typ = Funktionsbaustein (FB)
Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7513

CAN_RECOVER hat folgende Aufgaben:

- das automatische BusOff-Handling aktivieren / deaktivieren
- bei Vorliegen eines BusOff die CAN-Schnittstelle wieder neu starten.

> Wenn BusOff: CAN-Controller löscht alle Puffer (auch die Puffer der anderen Protokolle).

Wenn CAN_RECOVER nicht verwendet wird (ENABLE=FALSE):

> Bei einem BusOff wird automatisch nach 1 s ein Recover versucht.
> Nach 4 erfolglosen Recover-Versuchen in Folge wird die betroffene CAN-Schnittstelle deaktiviert.

Parameter der Eingänge

7514

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: KEIN automatisches Recover nach CAN-Busoff FALSE: Automatisches Recover nach CAN-Busoff
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
RECOVER_EXECUTE	BOOL	TRUE (nur 1 Zyklus lang): Neustart der CAN-Schnittstelle Busoff-Zustand beheben FALSE: Funktion wird nicht ausgeführt
InhibitTime (Parameter-Nutzung optional)	TIME := T#1s	Wartezeit zwischen BusOff und Neustart der CAN-Schnittstelle

© ifm electronic gmbh www.ifm.com

CAN_SETDOWNLOADID

7516

= Set Download-ID

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyyzz.LIB`

Symbol in CODESYS:

Beschreibung

7517

Die Download-ID ist zum Datenaustausch erforderlich bei der Verbindung zwischen dem Laufzeitsystem und der CODESYS-Entwicklungsumgebung. Die Download-ID wird voreingestellt beim Start des Geräts mit dem Wert aus der Hardware-Konfiguration gesetzt.

Mit CAN_SETDOWNLOADID kann dieser Wert im SPS-Programm (z.B. anhand bestimmter Eingänge) gesetzt werden. Die geänderte ID wird auch in die Hardware-Konfiguration geschrieben.

Parameter der Eingänge

7519

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DOWNLOAD_ID	BYTE	1...127 = Download-ID setzen 0 = Download-ID lesen

Parameter der Ausgänge

7520

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

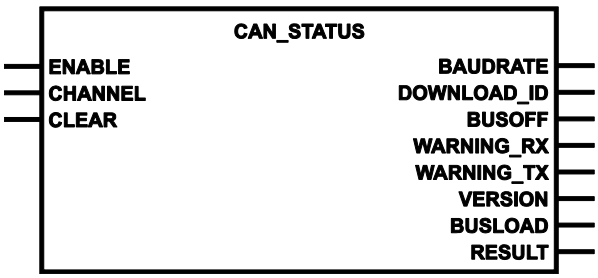
CAN_STATUS

7499

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7501

Mit CAN_STATUS können Informationen zum gewählten CAN-Bus abgefragt werden.

Ohne Hardware-Initialisierung können folgende Merker wieder auf FALSE gesetzt werden:
- BUSOFF
- WARNING_RX
- WARNING_TX

Parameter der Eingänge

7502

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
CLEAR	BOOL := FALSE	TRUE: Folgende Anzeigen zurücksetzen: • WARNING_RX • WARNING_TX • BUSOFF FALSE: Funktion wird nicht ausgeführt

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7504

Parameter	Datentyp	Beschreibung
BAUDRATE	WORD	aktuelle Baudrate des CANopen-Knotens in [kBaud]
DOWNLOAD_ID	BYTE	aktueller Download-ID
BUSOFF	BOOL	Fehler CAN-BUSOFF an der Schnittstelle
WARNING_RX	BOOL	Empfangs-Warnschwelle an der Schnittstelle überschritten
WARNING_TX	BOOL	Sende-Warnschwelle an der Schnittstelle überschritten
VERSION	DWORD	Version der ifm-CAN-Stack-Bibliothek
BUSLOAD	BYTE	mittlere Buslast in [%] zulässig: 0...100
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
8	08	Funktionsbaustein ist aktiv
9	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

Bausteine: RAW-CAN Daten empfangen

Inhalt

CAN_RX 74
CAN_RX_ENH 75
CAN_RX_ENH_FIFO 77
CAN_RX_RANGE 79
CAN_RX_RANGE_FIFO 81

15050

© ifm electronic gmbh

CAN_RX

7586

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyyzz.LIB`

Symbol in CODESYS:

Beschreibung

7588

CAN_RX dient zum Empfang einer Nachricht.
Der FB beschränkt sich auf wenige Funktionen und hat nur geringen Speicherbedarf.

CAN_RX filtert auf den eingestellten Identifier. Wenn innerhalb eines Zyklus mehrere CAN-Nachrichten mit dem gleichen Identifier empfangen werden, steht nur die letzte / aktuellste Nachricht zur Verfügung.

Parameter der Eingänge

7589

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifiers: Normal Frame (2^{11} IDs): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2^{29} IDs): 2 048...536 870 911 = 0x0000 0800...0x1FFF FFFF

Parameter der Ausgänge

7590

Parameter	Datentyp	Beschreibung
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist in der Bearbeitung – Empfangen läuft noch
9	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

CAN_RX_ENH

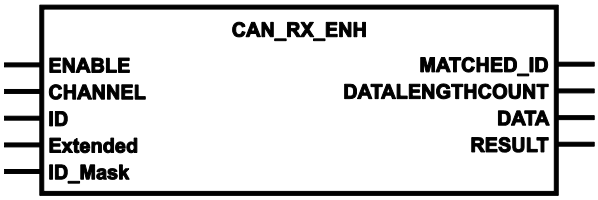
7606

= CAN RX enhanced

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7608

CAN_RX_ENH bietet (im Vergleich zu ***CAN_RX*** (→ Seite 74)) zusätzlich folgende Möglichkeiten:
- den Frame-Typ wählen (11 oder 29 Bit),
- eine Maske für die Auswertung des CAN-ID definieren.

Bit-Vergleich von ID und Maske:	Wenn ID_MASK-Bit = 0, dann darf CAN-ID-Bit = 0 oder 1 sein. Wenn ID_MASK-Bit = 1, dann muss CAN-ID-Bit = ID-Bit sein.

Mit der Maske können Sie mehrere Identifier als Filter definieren.

Beispiel:

ID =	0x100 = 0b0001 0000 0000
ID_MASK =	0x1F1 = 0b0001 1111 0001
Ergebnis	Die CAN-IDs mit folgendem Bitmuster werden ausgewertet: 0bxxx1 0000 xxx0 (x = beliebig), also für dieses Beispiel (alles in [hex]): 100, 102, 104, 106, 108, 10A, 10C, 10E, 300, 302, 304, 306, 308, 30A, 30C, 30E, 500, 502, 504, 506, 508, 50A, 50C, 50E, 700, 702, 704, 706, 708, 70A, 70C, 70E

Parameter der Eingänge

7609

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifiers: Normal Frame (2^{11} IDs): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2^{29} IDs): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Extended Frame (ID = 0...2^{29}-1) FALSE: Normal Frame (ID = 0...2^{11}-1)
ID_Mask (Parameter-Nutzung optional)	DWORD := 0	Filter-Maskierung zum Identifier: Wenn ID_MASK-Bit = 0, dann darf CAN-ID-Bit = 0 oder 1 sein. Wenn ID_MASK-Bit = 1, dann muss CAN-ID-Bit = ID-Bit sein.

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7613

Parameter	Datentyp	Beschreibung
MATCHED_ID	DWORD	Nummer des Datenobjekt-Identifiers
DATALENGTHCOUNT	BYTE	= Data Length Count Anzahl der empfangenen Daten-Bytes
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist in der Bearbeitung – Empfangen läuft noch
9	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

CAN_RX_ENH_FIFO

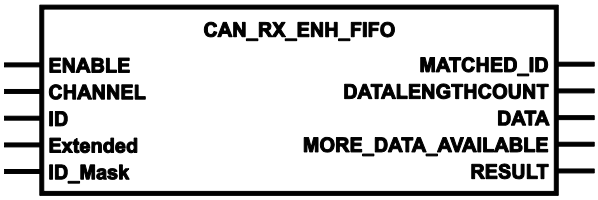
7615

= CAN RX enhanced with FIFO

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7616

CAN_RX_ENH_FIFO stellt (im Vergleich zu ***CAN_RX_ENH*** (→ Seite 75)) zusätzlich ein FiFo für die empfangenen Daten zur Verfügung. Somit können mehrere CAN-Telegramme innerhalb eines Zyklus empfangen werden.

> Wenn das FiFo voll ist, wird nicht überschrieben. Eingehende Nachrichten gehen dann verloren.

In diesem Fall:

► Mittels ENABLE den FB deaktivieren und wieder aktivieren.

> Das FiFo wird gelöscht und kann von neuem befüllt werden.

Parameter der Eingänge

7609

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifiers: Normal Frame (2^{11} IDs): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2^{29} IDs): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Extended Frame (ID = 0...2^{29}-1) FALSE: Normal Frame (ID = 0...2^{11}-1)
ID_Mask (Parameter-Nutzung optional)	DWORD := 0	Filter-Maskierung zum Identifier: Wenn ID_MASK-Bit = 0, dann darf CAN-ID-Bit = 0 oder 1 sein. Wenn ID_MASK-Bit = 1, dann muss CAN-ID-Bit = ID-Bit sein.

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7617

Parameter	Datentyp	Beschreibung
MATCHED_ID	DWORD	Nummer des Datenobjekt-Identifiers
DATALENGTHCOUNT	BYTE	= Data Length Count Anzahl der empfangenen Daten-Bytes
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
MORE_DATA_AVAILABLE	BOOL	TRUE: weitere empfangene Daten im FiFo vorhanden FALSE: keine weiteren Daten im FiFo vorhanden
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist in der Bearbeitung – Empfangen läuft noch
9	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren

© ifm electronic gmbh www.ifm.com

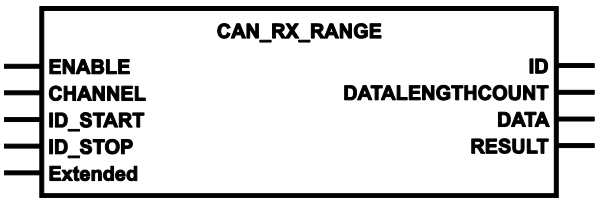
CAN_RX_RANGE

7592

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7594

CAN_RX_RANGE bietet folgende Einstellungen:

- den Nachrichten-Typ wählen (11 oder 29 Bit),
- einen Identifier-Bereich definieren.

CAN_RX filtert auf den eingestellten Identifier. Wenn innerhalb eines Zyklus mehrere CAN-Nachrichten mit dem gleichen Identifier empfangen werden, steht nur die letzte / aktuellste Nachricht zur Verfügung.

Parameter der Eingänge

7595

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID_START	DWORD	Anfangs-Nummer des Datenobjekt-Identifier-Bereichs: Normal Frame (2^{11}): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2^{29}): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
ID_STOP	DWORD	End-Nummer des Datenobjekt-Identifier-Bereichs: Normal Frame (2^{11}): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2^{29}): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Extended Frame (ID = 0...2^{29}-1) FALSE: Normal Frame (ID = 0...2^{11}-1)

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7598

Parameter	Datentyp	Beschreibung
ID	DWORD	Nummer des Datenobjekt-Identifiers
DATALENGTHCOUNT	BYTE	= Data Length Count Anzahl der empfangenen Daten-Bytes
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist in der Bearbeitung – Empfangen läuft noch
9	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

CAN_RX_RANGE_FIFO

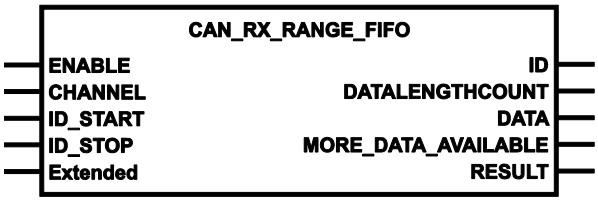
7601

= CAN RX Range with FIFO

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7603

CAN_RX_RANGE_FIFO arbeitet grundsätzlich wie ***CAN_RX_RANGE*** (→ Seite 79).

Zusätzlich stellt CAN_RX_RANGE_FIFO ein FiFo für die empfangenen Daten zur Verfügung. Somit können mehrere CAN-Telegramme innerhalb eines Zyklus empfangen werden.

Wenn das FiFo voll ist, wird nicht überschrieben. Eingehende Nachrichten gehen dann verloren.

In diesem Fall:

► Mit ENABLE die Funktion deaktivieren und wieder aktivieren.

> Das FiFo wird gelöscht und kann von neuem befüllt werden.

Parameter der Eingänge

7595

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID_START	DWORD	Anfangs-Nummer des Datenobjekt-Identifier-Bereichs: Normal Frame (2^{11}): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2^{29}): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
ID_STOP	DWORD	End-Nummer des Datenobjekt-Identifier-Bereichs: Normal Frame (2^{11}): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2^{29}): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Extended Frame (ID = 0...2^{29}-1) FALSE: Normal Frame (ID = 0...2^{11}-1)

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7604

Parameter	Datentyp	Beschreibung
ID	DWORD	Nummer des Datenobjekt-Identifiers
DATALENGTHCOUNT	BYTE	= Data Length Count Anzahl der empfangenen Daten-Bytes
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
MORE_DATA_AVAILABLE	BOOL	TRUE: weitere empfangene Daten im FiFo vorhanden FALSE: keine weiteren Daten im FiFo vorhanden
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist in der Bearbeitung – Empfangen läuft noch
9	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren

© ifm electronic gmbh www.ifm.com

Bausteine: RAW-CAN Daten senden

Inhalt

CAN_TX 84
CAN_TX_ENH 85
CAN_TX_ENH_CYCLIC 87

15055

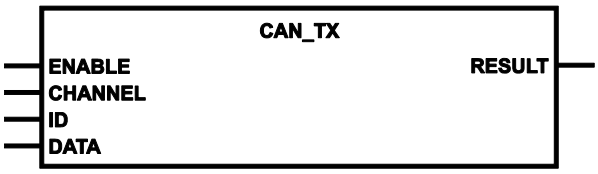
© ifm electronic gmbh

CAN_TX

7522

Baustein-Typ = Funktionsbaustein (FB)
Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7523

CAN_TX sendet eine Standard-Nachricht pro Zyklus.
Der FB beschränkt sich auf wenige Funktionen und hat nur geringen Speicherbedarf.

> Bei mehrmaligem Aufruf derselben Instanz dieses FBs während eines Zyklus werden die Daten ebenfalls mehrmals versendet.

Bei den einfachen Funktionen CAN_TX und CAN_RX wird anhand des ID ermittelt, ob ein Standard- oder ein Extended-Frame versendet werden soll. Bei den Enhanced-Versionen wird dies über den Eingang EXTENDED festgelegt. Mit den einfachen Funktionen kann man folglich keine Extended-Frames im ID-Bereich 0...2047 versenden.

Parameter der Eingänge

7524

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifiers: Normal Frame (2^{11} IDs): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2^{29} IDs): 2 048...536 870 911 = 0x0000 0800...0x1FFF FFFF
DATA	ARRAY [0..7] OF BYTE	zu sendende Daten (1...8 Bytes)

Parameter der Ausgänge

7527

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren

© ifm electronic gmbh www.ifm.com

CAN_TX_ENH

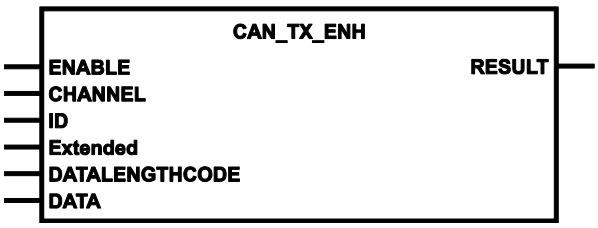
7558

= CAN TX enhanced

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7559

Zusätzliche Einstellmöglichkeiten bietet CAN_TX_ENH (für: enhanced). Hier können alle CAN-spezifischen Eigenschaften individuell eingestellt werden, z.B.:

- Handelt es sich um einen 11- oder 29-Bit-Identifier?
- Die zusätzlichen Eingänge können voreingestellt werden, so dass ***CAN_TX*** (→ Seite 84) nicht erforderlich ist.

> Bei mehrmaligem Aufruf derselben Instanz dieses FBs während eines Zyklus werden die Daten ebenfalls mehrmals versendet.

Parameter der Eingänge

7634

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein initialisieren (nur 1 Zyklus) > Baustein-Eingänge lesen TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifiers: Normal Frame (2^{11} IDs): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2^{29} IDs): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Extended Frame (ID = 0...2^{29}-1) FALSE: Normal Frame (ID = 0...2^{11}-1)
DATALENGTHCODE	BYTE	= Data Length Code Anzahl der zu sendenden Daten-Bytes (0...8)
DATA	ARRAY [0..7] OF BYTE	zu sendende Daten (1...8 Bytes)

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7527

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren

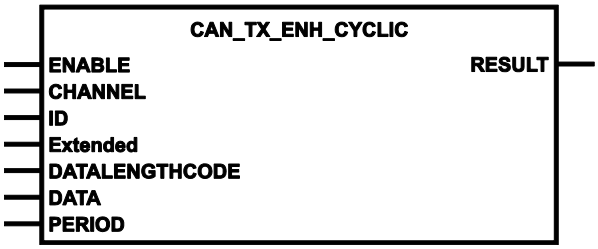
© ifm electronic gmbh www.ifm.com

CAN_TX_ENH_CYCLIC

7568

= CAN TX enhanced Cyclic
Baustein-Typ = Funktionsbaustein (FB)
Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7569

CAN_TX_ENH_CYCLIC dient dem zyklischen Versand von CAN-Nachrichten.
Der FB entspricht ansonsten ***CAN_TX_ENH*** (→ Seite 85).

► Mit dem Parameter PERIOD die Periodendauer einstellen.

Eine zu kurze Periodendauer kann zu einer hohen Buslast führen, was das Verhalten des Gesamtsystems beinträchtigen könnte.

Parameter der Eingänge

7582

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifiers: Normal Frame (2^{11} IDs): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2^{29} IDs): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Extended Frame (ID = 0...2^{29}-1) FALSE: Normal Frame (ID = 0...2^{11}-1)
DataLengthCode (Parameter-Nutzung optional)	BYTE := 8	Länge der zu sendenden Daten (0...8 Bytes)
DATA	ARRAY [0..7] OF BYTE	zu sendende Daten (1...8 Bytes)
PERIOD	TIME	Periodendauer

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7510

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
8	08	Funktionsbaustein ist aktiv
9	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

Bausteine: RAW-CAN Remote

Inhalt

CAN_REMOTE_REQUEST 90
CAN_REMOTE_RESPONSE 91

15057

© ifm electronic gmbh www.ifm.com

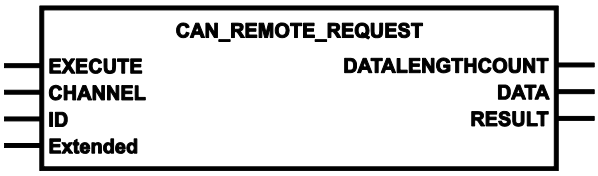
CAN_REMOTE_REQUEST

7625

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7627

Zur Anfrage einer Remote-Nachricht wird mit CAN_REMOTE_REQUEST eine entsprechende Anforderung versandt und die Antwort des anderen Gerätes als Ergebnis zurückgeliefert.

Parameter der Eingänge

7628

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifiers: Normal Frame (2^{11} IDs): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2^{29} IDs): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Extended Frame (ID = 0...2^{29}-1) FALSE: Normal Frame (ID = 0...2^{11}-1)

Parameter der Ausgänge

7629

Parameter	Datentyp	Beschreibung
DATALENGTHCOUNT	BYTE	= Data Length Count Anzahl der empfangenen Daten-Bytes
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist in der Bearbeitung – Empfangen läuft noch
9	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

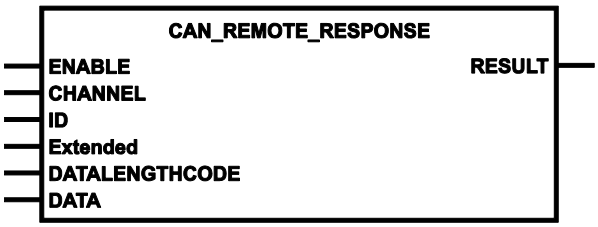
CAN_REMOTE_RESPONSE

7631

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_RawCAN_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7633

CAN_REMOTE_RESPONSE stellt dem CAN-Controller im Gerät Daten zur Verfügung, die automatisch auf die Anfrage einer Remote-Nachricht gesendet werden.

Dieser FB ist stark geräte-abhängig. Es kann nur eine begrenzte Anzahl von Remote-Nachrichten eingerichtet werden:

BasicController: CR040n, CR041n, CR043n BasicDisplay: CR045n	max. 40 Remote-Nachrichten
PDM360 NG: CR108n	max. 100 Remote-Nachrichten

Parameter der Eingänge

7634

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein initialisieren (nur 1 Zyklus) > Baustein-Eingänge lesen TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ID	DWORD	Nummer des Datenobjekt-Identifiers: Normal Frame (2^{11} IDs): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2^{29} IDs): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
Extended (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Extended Frame (ID = 0...2^{29}-1) FALSE: Normal Frame (ID = 0...2^{11}-1)
DATALENGTHCODE	BYTE	= Data Length Code Anzahl der zu sendenden Daten-Bytes (0...8)
DATA	ARRAY [0..7] OF BYTE	zu sendende Daten (1...8 Bytes)

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7636

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
6	06	FB ist in der Bearbeitung – Remote für ID nicht aktiv
7	07	FB ist in der Bearbeitung – Remote für ID aktiv

© ifm electronic gmbh www.ifm.com

5.2.3 Bausteine: CANopen

Inhalt

Bausteine: CANopen Status ... 93
Bausteine: CANopen Netzwerkmanagement ... 102
Bausteine: CANopen Objektverzeichnis ... 106
Bausteine: CANopen SDOs ... 111
Bausteine: CANopen SYNC ... 124
Bausteine: CANopen Guarding ... 128
Bausteine: CANopen Emergency ... 132

15059

Für CANopen stellt **ifm electronic** eine Reihe von Bausteinen zur Verfügung, die im Folgenden erklärt werden.

Bausteine: CANopen Status

Inhalt

CANOPEN_ENABLE ... 94
CANOPEN_GETBUFFERFLAGS ... 96
CANOPEN_GETSTATE ... 98
CANOPEN_SETSTATE ... 100

15061

© ifm electronic gmbh

CANOPEN_ENABLE

7785

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:

Beschreibung

7787

CANOPEN_ENABLE erlaubt es, den CANopen-Master ein- und auszuschalten.

► Im Anwendungsprogramm für jede CAN-Schnittstelle immer eine eigene Instanz des FBs ***CANOPEN_ENABLE*** (→ Seite 94) anlegen!

Zur Vermeidung von Guarding- oder Heartbeat-Fehlern müssen zuvor die Knoten durch eine geeignete Sequenz "heruntergefahren" werden.

Wird der Master nach einem Stopp wieder gestartet, so müssen auch alle angeschlossenen Knoten wieder initialisiert werden.

Ohne CANOPEN_ENABLE wird der CANopen-Master automatisch gestartet, sofern dies in der Konfiguration gewählt wurde.

Die konfigurierte Baudrate wird nur übernommen, wenn zuvor nicht ***CAN_ENABLE*** (→ Seite 68) aufgerufen wurde.

Parameter der Eingänge

7788

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := TRUE	TRUE: • CANopen für den gewählten Kanal freigeben • CANopen-Manager oder CANopen-Device starten entsprechend den Konfigurations-Einstellungen FALSE: • CANopen für den gewählten Kanal sperren • CANopen-Manager oder CANopen-Device beenden
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
Baudrate (Parameter-Nutzung optional)	WORD := 0	Baudrate [kBit/s] zulässig = 20, 50, 100, 125, 250, 500, 800, 1 000 0 = Einstellung aus der Steuerungskonfiguration verwenden

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7789

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
14	0E	FB ist aktiv CANopen-Manager konfiguriert Devices und sendet SDOs
15	0F	FB ist aktiv CANopen-Manager ist gestartet
238	EE	Fehler: CANopen-Konfiguration ist zu groß und kann nicht gestartet werden
239	EF	Fehler: CANopen-Manager konnte nicht gestartet werden
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

CANOPEN_GETBUFFERFLAGS

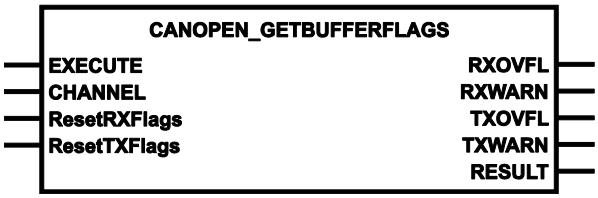
7890

= Get Buffer-Flags

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7892

CANOPEN_GETBUFFERFLAGS liefert Informationen zu den Buffer-Flags.
Über optionale Eingänge können die Flags zurückgesetzt werden.
Der Funktionsbaustein liefert den Zustand der Overflow-Flags zurück.

Parameter der Eingänge

7893

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ResetRXFlags (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Flag-Status am Ausgang ausgeben und anschließend zurücksetzen FALSE: Funktion wird nicht ausgeführt
ResetTXFlags (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Flag-Status am Ausgang ausgeben und anschließend zurücksetzen FALSE: Funktion wird nicht ausgeführt

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7894

Parameter	Datentyp	Beschreibung
RXOVFL	BOOL	Zustand des RX-Overflow-Flags TRUE: Überlauf im Empfangspuffer FALSE: Kein Überlauf im Empfangspuffer
RXWARN	BOOL	Zustand des RX-Overflow-Warning-Flags TRUE: Füllstand im Empfangspuffer ist kritisch FALSE: Füllstand im Empfangspuffer ist unkritisch
TXOVFL	BOOL	Zustand des TX-Overflow-Flags TRUE: Überlauf im Sendepuffer FALSE: Kein Überlauf im Sendepuffer
TXWARN	BOOL	Zustand des TX-Overflow-Warning-Flags TRUE: Füllstand im Sendepuffer ist kritisch FALSE: Füllstand im Sendepuffer ist unkritisch
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist noch nicht ausgeführt
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

CANOPEN_GETSTATE

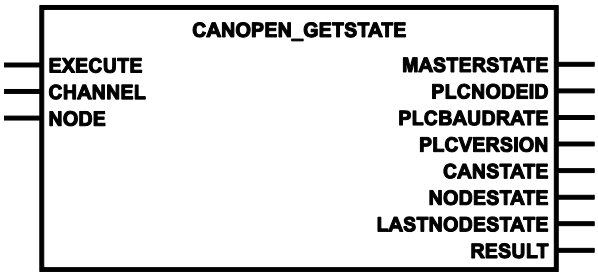
7865

= Get State

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7867

Mit CANOPEN_GETSTATE können Parameter des Masters, eines Slave-Devices oder eines bestimmten Knotens im Netz abgefragt werden.

Parameter der Eingänge

7868

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	Node-ID = ID des Knotens (0...127) **Gerät als CANopen-Master:** Wert = 0: Nur die Statusinformationen des Geräts selbst werden an den Ausgängen zurückgeliefert. Die Ausgänge mit Informationen zu den Knoten sind ungültig. Wert nicht 0: Node-ID eines Knotens im Netzwerk. Für diesen sowie für das Gerät werden an den Ausgängen die Zustände zurückgeliefert. **Gerät als CANopen-Slave:** Wert = 0 (voreingestellt): Die Statusinformationen des Slave werden an den Ausgängen zurückgeliefert. Wert nicht 0: keine Aktion

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7869

Parameter	Datentyp	Beschreibung
MASTERSTATE	BYTE	Master State = interner Zustand des Masters: 0 = 0x00 = Master läuft hoch 4 = 0x04 = Konfiguration der Knoten läuft 5 = 0x05 = normaler Betriebszustand des Masters 255 = 0xFF = PLC läuft als Slave
PLCNODEID	BYTE	PLC Node-ID = Node-ID der SPS, auf der das Programm ausgeführt wird Wert = 0...127 = 0x00...0x7F
PLCBAUDRATE	DWORD	Baudrate der SPS
PLCVERSION	DWORD	Version der SPS
CANSTATE	BYTE	Status des CANopen-Netzwerks **Gerät als Master betrieben:** **Node-ID = 0 (Gerät selbst):** 0 = 0x00 = OK 128 = 0x80 = BUSOFF **Node-ID ≠ 0 (Knoten):** 0 = 0x00 = OK 1 = 0x01 = Guard- oder Heartbeat-Fehler an Knoten 128 = 0x80 = BUSOFF **Gerät als Slave betrieben:** 0 = 0x00 = OK 1 = 0x01 = Guard- oder Heartbeat-Fehler 128 = 0x80 = BUSOFF
NODESTATE	BYTE	Node State = interner Knotenstatus eines Slaves aus Sicht des Masters. Der Knoten wird durch den Eingang NODEID bezeichnet. -1 = 0xFF = Rücksetzen nach ResetNode 1 = 0x01 = Warten auf BOOTUP 2 = 0x02 = Nach Empfang der BOOTUP-Nachricht 3 = 0x03 = noch nicht konfiguriert: STOPPED 4 = 0x04 = nach der Konfiguration mit SDOs: PRE-OPERATIONAL 5 = 0x05 = nach dem Starten des Knotens: OPERATIONAL 97 = 0x61 = optionaler Knoten 98 = 0x62 = anderer Gerätetyp als in 0x1000 konfiguriert 99 = 0x63 = Node-Guarding
LASTNODESTATE	BYTE	Last Node State = letzter Status des Knotens Knotenstatus nach CANopen (mit diesen Werten wird der Status auch in den entsprechenden Nachrichten vom Knoten her codiert). 0 0x00 BOOTUP 4 0x04 STOPPED 5 0x05 OPERATIONAL 127 0x7F PRE-OPERATIONAL
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
8	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh

CANOPEN_SETSTATE

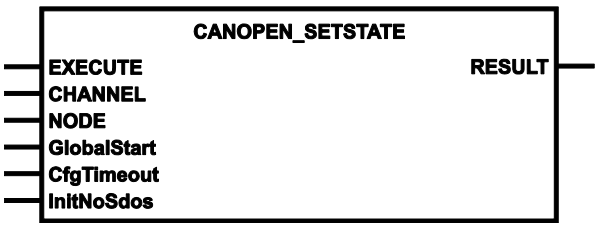
7858

= Set State

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7860

Mit CANOPEN_SETSTATE können Parameter des Masters, eines Slave-Devices oder eines Knotens im Netz gesetzt werden.

Die Behandlung des NMT-Zustands von Master, Knoten oder Device erfolgt im CAN-Stack oder über die Kommandos des FB ***CANOPEN_NMTSERVICES*** (→ Seite 104). Dabei werden gleichzeitig auch Zulässigkeitsprüfungen durchgeführt. Aus Konsistenzgründen sind deshalb hier für diesen Zweck keine Eingänge vorgesehen.

© ifm electronic gmbh www.ifm.com

Parameter der Eingänge

7861

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	Node-ID = ID des Knotens (0...127) **Gerät als CANopen-Master:** Wert = 0: Die Änderungen beziehen sich nur auf das Gerät selbst. Wert nicht 0: Node-ID eines Knotens am Netzwerk, dessen Parameter verändert werden sollen. Nur für diesen Knoten (nicht für das Gerät) werden die angelegten Einstellungen übernommen. **Gerät als CANopen-Slave:** Im Slave-Mode kann über diesen Eingang der Node-ID des Slave gesetzt werden. Wert = 0: keine Aktion Wert nicht 0: Diesen Wert übernimmt der FB als neuen Node-ID des Geräts.
GlobalStart (Parameter-Nutzung optional)	BOOL := TRUE	Voraussetzung: FB muss unmittelbar nach dem Start des IEC-Programms aufgerufen werden. Diese Einstellung überschreibt die Einstellung aus der Konfiguration. TRUE: alle Teilnehmer gleichzeitig starten FALSE: alle Teilnehmer nacheinander starten
CfgTimeout (Parameter-Nutzung optional)	TIME := T#0ms	Konfigurations-Timeout für einen Knoten setzen: Wert = 0: keine Aktion – Konfigurationsdaten behalten Wert nicht 0: Daten aus der Konfiguration mit dem neuen Wert überschreiben
InitNoSdos (Parameter-Nutzung optional)	BOOL := FALSE	An den in NODE angegebenen Knoten beim Initialisieren... TRUE: keine Konfigurationsdaten senden FALSE: die konfigurierten SDOs senden

Parameter der Ausgänge

7862

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
8	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh

Bausteine: CANopen Netzwerkmanagement

Inhalt

CANOPEN_GETNMTSTATESLAVE 103
CANOPEN_NMTSERVICES 104

15063

© ifm electronic gmbh www.ifm.com

CANOPEN_GETNMTSTATESLAVE

7851

= Get Network Management State Slave

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:

Beschreibung

7853

► Baustein nur verwenden, wenn das Gerät als CANopen-Slave betrieben wird!

An die Anwendung wird mit CANOPEN_GETNMTSTATESLAVE nur noch der Betriebszustand nach CANopen gemeldet sowie eine Fehlermeldung, falls ein ungültiger Zustandsübergang angefordert wurde.

Parameter der Eingänge

7854

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät

Parameter der Ausgänge

7855

Parameter	Datentyp	Beschreibung
NMTSTATE	BYTE	Netzwerk-Betriebszustand des Knotens 0 = INIT 1 = OPERATIONAL 2 = PRE-OPERATIONAL 3 = STOPPED
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

CANOPEN_NMTSERVICES

7843

= Network Management Services

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:

Beschreibung

7844

CANOPEN_NMTSERVICES löst abhängig von seinen NMT-Kommando-Eingängen ein NMT-Kommando oder die Initialisierung eines Knotens aus.

NMT = **N**etwork-**M**anagemen**T**

Der Funktionsbaustein aktualisiert den internen Knotenstatus. Sollte ein Zustandsübergang nach CANopen (→ ***NMT-Status***) nicht erlaubt sein, so wird das Kommando nicht ausgeführt.

Ein CANopen-Device kann mit Hilfe des FB seinen CANopen-Status selbständig ändern: Preoperational ⇦⇨ Operational

Parameter der Eingänge

7847

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	CANopen-ID des Knotens zulässig = 1...127 = 0x00...0x7F NODE = 0: Kommando gilt für alle Knoten im Netzwerk NODE = Node-ID des Geräts: Kommando gilt für das Gerät selbst
NMTSERVICE	BYTE	Netzwerk-Kommando 0 = Init Node (außer Master) 1 = Enter PRE-OPERATIONAL 2 = Start Node 3 = Reset Node 4 = Reset Kommunikation 5 = Stop Node
Timeout (Parameter-Nutzung optional)	TIME := T#0ms	Wartezeit des FB auf die Initialisierung Nach Zeitablauf bricht der FB das Warten ab. 0 = Wert aus der Konfiguration verwenden

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7848

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist aktiv
35	23	mindestens 1 SDO der Konfiguration war nicht erfolgreich
36	24	Knoten war bereits initialisiert
37	25	zur Initialisierung war Knoten nicht im Modus PRE-OPERATIONAL
043	2B	Master / Slave ist nicht initialisiert
241	F1	Fehler: CANopen-Zustandsübergang ist nicht erlaubt
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

Bausteine: CANopen Objektverzeichnis

Inhalt

CANOPEN_GETODCHANGEDFLAG 107
CANOPEN_READOBJECTDICT 108
CANOPEN_WRITEOBJECTDICT 109

15065

© ifm electronic gmbh www.ifm.com

CANOPEN_GETODCHANGEDFLAG

7927

= Get Object Directory Changed Flag

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:

Beschreibung

7928

CANOPEN_GETODCHANGEDFLAG meldet wenn bei einem bestimmten Objektverzeichnis-Eintrag der Wert geändert wurde.

Parameter der Eingänge

7930

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis

Parameter der Ausgänge

7931

Parameter	Datentyp	Beschreibung
DATA	DWORD	Parameter-Wert
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

CANOPEN_READOBJECTDICT

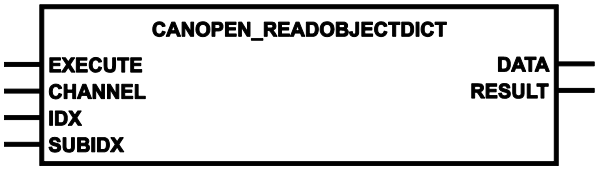
7933

= Read Object Directory

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7935

CANOPEN_READOBJECTDICT liest bis zu 4 Bytes Konfigurationsdaten aus dem Objektverzeichnis des Geräts zur Verwendung im Anwendungsprogramm.

Parameter der Eingänge

7936

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis

Parameter der Ausgänge

7937

Parameter	Datentyp	Beschreibung
DATA	DWORD	Parameter-Wert
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist noch nicht ausgeführt
40	28	Objektverzeichnis-Eintrag ist ungültig
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

CANOPEN_WRITEOBJECTDICT

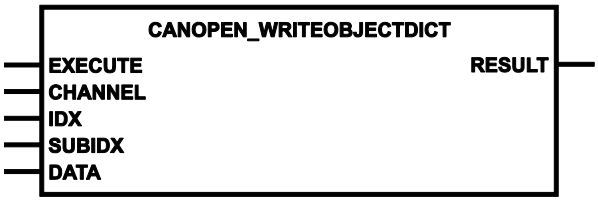
7940

= Write Object Directory

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7942

CANOPEN_WRITEOBJECTDICT schreibt Konfigurationsdaten in das Objektverzeichnis der Steuerung.

ACHTUNG

Wichtige Systemeinstellungen können hierbei verfälscht werden, z.B.:
- Guarding-Zeiten
- Heartbeat-Zeiten

► Eingabe-Parameter sorgfältig prüfen!

Parameter der Eingänge

7943

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis
DATA	DWORD	Parameter-Wert

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7945

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist noch nicht ausgeführt
40	28	Objektverzeichnis-Eintrag ist ungültig
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

Bausteine: CANopen SDOs

Inhalt

CANOPEN_SDOREAD 112
CANOPEN_SDOREADBLOCK 114
CANOPEN_SDOREADMULTI 116
CANOPEN_SDOWRITE 118
CANOPEN_SDOWRITEBLOCK 120
CANOPEN_SDOWRITEMULTI 122

2071

Hier finden Sie ifm-Bausteine für den Umgang von CANopen mit Service Data Objects (SDOs).

© ifm electronic gmbh

CANOPEN_SDOREAD

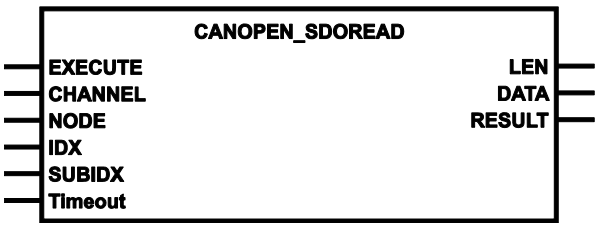
7791

= SDO Read

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7793

CANOPEN_SDOREAD ist ein einfacher Funktionsbaustein zur Bearbeitung von "Expedited SDOs", also SDOs mit maximal 4 Nutzdaten-Bytes. Diese Art bildet in der Regel einen Großteil der SDO-Kommunikation ab.

Expedited SDO = beschleunigtes Nachrichten-Objekt mit Servicedaten

Wegen der auf max. 4 Nutzdaten-Bytes begrenzten Datenmenge lässt sich erheblich Speicherplatz sparen, da dieser FB nur 4 Bytes als Pufferspeicher vorhalten muss und selbst kein großes Daten-Array anlegt.

Parameter der Eingänge

7794

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	CANopen-ID des Knotens zulässig = 1...127 = 0x01...0x7F
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis
Timeout (Parameter-Nutzung optional)	TIME := T#10ms	Wartezeit des FB auf die Antwort Nach Zeitablauf bricht der FB das Warten ab. Wert = 0: Wert aus der Konfiguration verwenden

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7795

Parameter	Datentyp	Beschreibung
LEN	BYTE	Anzahl der empfangenen Bytes (1...4)
DATA	DWORD	der empfangene Datenwert
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist aktiv – noch keine Daten empfangen
32	20	SDO-Übertragung abgebrochen von Client oder Server (SDO-Abort-Code 0x80)
33	21	TIMEOUT ist abgelaufen
242	F2	Fehler: Einstellung ist nicht möglich
255	FF	Pufferüberlauf – zu viele Daten-Bytes wurden empfangen

© ifm electronic gmbh www.ifm.com

CANOPEN_SDOREADBLOCK

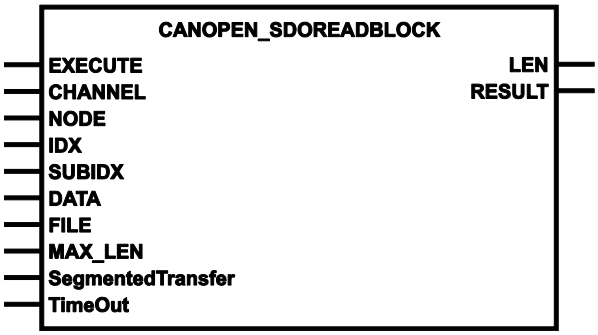
14942

= SDO Read Block

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

14943

CANOPEN_SDOREADBLOCK liest den angegeben Eintrag im Objektverzeichnis eines Knotens im Netz per SDO-Blocktransfer.

> Falls der Knoten keinen Blocktransfer unterstützt, schaltet der FB automatisch um auf den "Segmented Transfer". Per Eingang kann aber auch direkt auf den "Segmented Transfer" umgeschaltet werden.

> Die COB-ID für den SDO wird aus der übergebenen Node-ID berechnet.

Multiframe-SDOs sind in der Länge grundsätzlich nicht begrenzt.

Für Systeme ohne File-System (z.B. BasicController CR04nn) gilt:

► Dem FB eine Adresse übergeben, auf die per Zeiger schreibend zugegriffen wird. Der durch Startadresse DATA und Datenanzahl MAX_LEN definierte Speicherbereich muss verfügbar sein!

> Ist die Datenmenge größer als angegeben, wird der Transfer abgebrochen und per RESULT signalisiert.

Für Systeme mit File-System (z.B. PDM360NG CR108n) gilt:

► Dem FB den Pfad und Namen einer Datei übergeben, in welcher die Daten im Binärformat gespeichert werden sollen.

> Zum Status der SDO-Übertragung informiert der Ausgang RESULT.

© ifm electronic gmbh www.ifm.com

Parameter der Eingänge

14945

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	(Node-ID) ID des Knotens zulässig = 0x01...0x7F = 1...127 Die COB-ID des SDOs errechnet sich aus Knoten-ID + 0x600
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis
DATA	DWORD	Adresse des Datenbereichs zum Speichern der empfangenen Daten Eingang ist ohne Funktion bei Gerät mit Dateisystem (Linux).
FILE	STRING(80)	Pfad und Dateiname zum Speichern der empfangenen Daten im Binärformat Eingang ist ohne Funktion bei Gerät ohne Dateisystem (BasicSystem).
MAX_LEN	DWORD	Maximal erlaubte Anzahl der Bytes, die empfangen werden dürfen
SegmentedTransfer (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Segmented SDO-Transfer FALSE: SDO-Blocktransfer
Timeout (Parameter-Nutzung optional)	TIME := T#10ms	Wartezeit des FB auf die Antwort Nach Zeitablauf bricht der FB das Warten ab. Wert = 0: Wert aus der Konfiguration verwenden

Parameter der Ausgänge

14951

Parameter	Datentyp	Beschreibung
LEN	DWORD	Anzahl der empfangenen Daten-Bytes
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
16	10	Übertragung läuft als segmentierter Download
17	11	Übertragung läuft als Block-Download
32	20	SDO-Übertragung abgebrochen von Client oder Server (SDO-Abort-Code 0x80)
33	21	TIMEOUT ist abgelaufen
64	40	Fehler: Schreibzeiger ist außerhalb des zulässigen Datenbereichs
65	41	Fehler: Datei konnte nicht geöffnet werden
66	42	Fehler bei Schreiben auf Datei
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh

CANOPEN_SDOREADMULTI

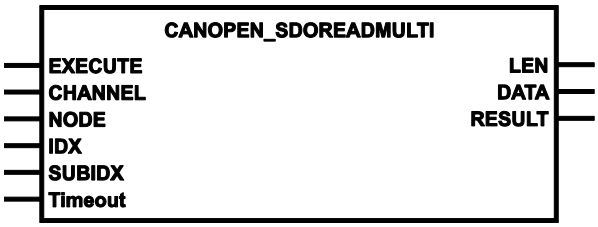
7806

= SDO Read Multi

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7808

CANOPEN_SDOREADMULTI liest den angegeben Eintrag im Objektverzeichnis eines Knotens im Netz. Die COB-ID für das SDO wird nach CANopen-Konvention aus der übergebenen Node-ID berechnet.

Parameter der Eingänge

7809

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	(Node-ID) ID des Knotens zulässig = 0x01...0x7F = 1...127 Die COB-ID des SDOs errechnet sich aus Knoten-ID + 0x600
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis
Timeout (Parameter-Nutzung optional)	TIME := T#10ms	Wartezeit des FB auf die Antwort Nach Zeitablauf bricht der FB das Warten ab. Wert = 0: Wert aus der Konfiguration verwenden

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7810

Parameter	Datentyp	Beschreibung
LEN	DWORD	Anzahl der empfangenen Bytes zulässige Werte = 0x0000 0001...0x0000 0800 = 1...2 048
DATA	ARRAY [0..SDOMAXDATA] OF BYTE	Pufferspeicher für Nutzdaten der SDO-Datenübertragung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist aktiv – noch keine Daten empfangen
32	20	SDO-Übertragung abgebrochen von Client oder Server (SDO-Abort-Code 0x80)
33	21	TIMEOUT ist abgelaufen
34	22	Toggle-Bit-Fehler
242	F2	Fehler: Einstellung ist nicht möglich
255	FF	Fehler: zu wenig Speicher für Empfangs-Multiframe verfügbar

© ifm electronic gmbh www.ifm.com

CANOPEN_SDOWRITE

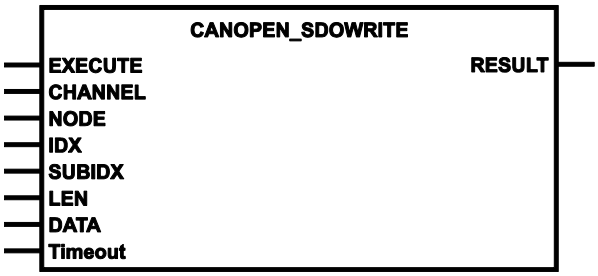
7825

= SDO Write

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7826

CANOPEN_SDOWRITE ist ein einfacher Funktionsbaustein zur Bearbeitung von "Expedited SDOs", also SDOs mit maximal 4 Nutzdaten-Bytes. Diese Art bildet in der Regel einen Großteil der SDO-Kommunikation ab.

Expedited SDO = beschleunigtes Nachrichten-Objekt mit Servicedaten

Wegen der auf max. 4 Nutzdaten-Bytes begrenzten Datenmenge lässt sich erheblich Speicherplatz sparen, da dieser FB nur 4 Bytes als Pufferspeicher vorhalten muss und selbst kein großes Daten-Array anlegt.

Parameter der Eingänge

7828

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	CANopen-ID des Knotens zulässig = 1...127 = 0x01...0x7F
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis
LEN	BYTE	Anzahl der zu übertragenden Daten-Bytes zulässige Werte = 1...4 = 0x01...0x04
DATA	ARRAY [0..3] OF BYTE	Datenbereich (1...4 Bytes)
Timeout (Parameter-Nutzung optional)	TIME := T#10ms	Wartezeit des FB auf die Antwort Nach Zeitablauf bricht der FB das Warten ab. Wert = 0: Wert aus der Konfiguration verwenden

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7829

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
8	08	Funktionsbaustein ist aktiv
32	20	SDO-Übertragung abgebrochen von Client oder Server (SDO-Abort-Code 0x80)
33	21	TIMEOUT ist abgelaufen
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

CANOPEN_SDOWRITEBLOCK

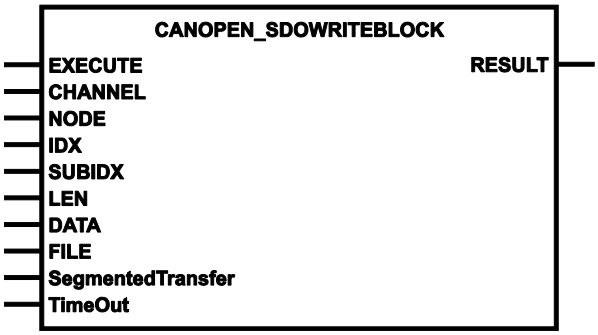
14961

= SDO Write Block

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

14963

CANOPEN_SDOWRITEBLOCK schreibt in den angegeben Eintrag im Objektverzeichnis eines Knotens im Netz per SDO-Blocktransfer.

Per FB-Eingang kann bei Bedarf auf den Segmented Transfer umgeschaltet werden.

> Die COB-ID für den SDO wird aus der übergebenen Node-ID berechnet.

> Zum Status der SDO-Übertragung informiert der Ausgang RESULT.

Multiframe-SDOs sind in der Länge grundsätzlich nicht begrenzt.

Für Systeme ohne File-System (z.B. BasicController CR04nn) gilt:

► Dem FB eine Adresse übergeben, auf die per Zeiger lesend zugegriffen wird.

Für Systeme mit File-System (z.B. PDM360NG CR108n) gilt:

► Dem FB den Pfad und Namen einer Datei übergeben, aus welcher die Daten im Binärformat gelesen werden sollen.

© ifm electronic gmbh www.ifm.com

Parameter der Eingänge

14964

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	(Node-ID) ID des Knotens zulässig = 0x01...0x7F = 1...127 Die COB-ID des SDOs errechnet sich aus Knoten-ID + 0x600
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis
LEN	DWORD	Anzahl der in DATA zu übergebenen Daten-Bytes zulässig = 1...2 048 = 0x0000 0001...0x0000 0800
DATA	DWORD	Adresse des Datenbereichs zum Lesen der zu sendenden Daten Eingang ist ohne Funktion bei Gerät mit Dateisystem (Linux).
FILE	STRING(80)	Pfad und Dateiname zum Lesen der zu sendenden Daten im Binärformat Eingang ist ohne Funktion bei Gerät ohne Dateisystem (BasicSystem).
SegmentedTransfer (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Segmented SDO-Transfer FALSE: SDO-Blocktransfer
Timeout (Parameter-Nutzung optional)	TIME := T#10ms	Wartezeit des FB auf die Antwort Nach Zeitablauf bricht der FB das Warten ab. Wert = 0: Wert aus der Konfiguration verwenden

Parameter der Ausgänge

14968

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
16	10	Übertragung läuft als segmentierter Download
17	11	Übertragung läuft als Block-Download
32	20	SDO-Übertragung abgebrochen von Client oder Server (SDO-Abort-Code 0x80)
33	21	TIMEOUT ist abgelaufen
65	41	Fehler: Datei konnte nicht geöffnet werden
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh

CANOPEN_SDOWRITEMULTI

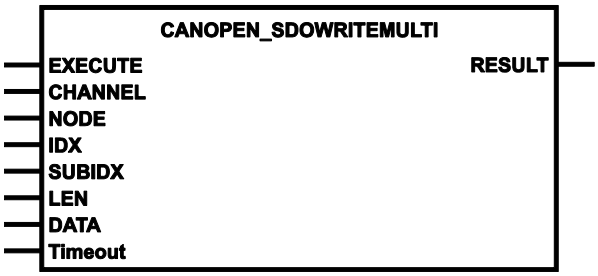
7832

= SDO Write Multi

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7834

CANOPEN_SDOWRITEMULTI schreibt den angegeben Eintrag im Objektverzeichnis eines Knotens im Netz. Die COB-ID für den SDO wird nach CANopen-Konvention aus der übergebenen Node-ID berechnet.

Parameter der Eingänge

7835

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
NODE	BYTE	CANopen-ID des Knotens zulässig = 1...127 = 0x01...0x7F
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis
LEN	DWORD	Anzahl der zu übertragenden Daten-Bytes zulässige Werte = 0000 0001...0000 0800_{16} = 1...2 048_{10}
DATA	ARRAY [0..SDOMAXDATA] OF BYTE	Pufferspeicher für Nutzdaten der SDO-Datenübertragung
Timeout (Parameter-Nutzung optional)	TIME := T#10ms	Wartezeit des FB auf die Antwort Nach Zeitablauf bricht der FB das Warten ab. Wert = 0: Wert aus der Konfiguration verwenden

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7836

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
8	08	Funktionsbaustein ist aktiv
32	20	SDO-Übertragung abgebrochen von Client oder Server (SDO-Abort-Code 0x80)
33	21	TIMEOUT ist abgelaufen
34	22	Toggle-Bit-Fehler
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

Bausteine: CANopen SYNC

Inhalt

CANOPEN_GETSYNCSTATE 125
CANOPEN_SETSYNCSTATE 127

15069

© ifm electronic gmbh www.ifm.com

CANOPEN_GETSYNCSTATE

7871

= Get SYNC State

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:

Beschreibung

7872

CANOPEN_GETSYNCSTATE liest...
- die Einstellung der SYNC-Funktionalität (aktiv / deaktiv),
- den Fehlerzustand der SYNC-Funktionalität (SyncError).

Wenn die PLC als CANopen-Slave läuft, wird über diesen FB signalisiert, ob SYNC-Signale ausbleiben oder ob sie regelmäßig kommen.

Die Bearbeitung von synchronen PDOs usw. läuft im CAN-Stack. CANOPEN_GETSYNCSTATE liefert jedoch den Fehlerzustand, so dass das Anwendungsprogramm darauf entsprechend reagieren kann.

Parameter der Eingänge

7874

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7875

Parameter	Datentyp	Beschreibung
SYNC	BOOL	Status der SYNC-Funktionalität TRUE: SYNC ist aktiviert: Im **Master-Betrieb** werden SYNC-Telegramme erzeugt entsprechend Einstellungen in Konfiguration sowie synchrone PDOs gesendet und empfangen. Im **Slave-Betrieb** werden SYNC-Telegramme empfangen und entsprechend bearbeitet. FALSE: SYNC ist nicht aktiv
SYNCERROR	BYTE	(Sync-Error) SYNC-Fehlermeldung 0 = kein Fehler >0 = SYNC-Error (Slave-Betrieb)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist noch nicht ausgeführt
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh

CANOPEN_SETSYNCSTATE

7883

= Set SYNC State

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:

Beschreibung

7884

Mit CANOPEN_SETSYNCSTATE wird die SYNC-Funktionalität ein- und ausgeschaltet.

Parameter der Eingänge

7886

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
SYNC	BOOL	Status der SYNC-Funktionalität TRUE: SYNC ist aktiviert: Im **Master-Betrieb** werden SYNC-Telegramme erzeugt entsprechend Einstellungen in Konfiguration sowie synchrone PDOs gesendet und empfangen. Im **Slave-Betrieb** werden SYNC-Telegramme empfangen und entsprechend bearbeitet. FALSE: SYNC ist nicht aktiv

Parameter der Ausgänge

7887

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist noch nicht ausgeführt
38	26	SYNC konnte nicht aktiviert werden
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

Bausteine: CANopen Guarding

Inhalt

CANOPEN_GETGUARDHBERRLIST 129
CANOPEN_GETGUARDHBSTATSLV 130

15071

© ifm electronic gmbh www.ifm.com

CANOPEN_GETGUARDHBERRLIST

7896

= Get Guard and Heartbeat Error-List

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:

Beschreibung

7898

CANOPEN_GETGUARDHBERRLIST listet in einem Array alle Knoten auf, für die der Master einen Fehler erkannt hat:
- Guarding-Fehler
- Heartbeat-Fehler

Parameter der Eingänge

7899

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ResetList (Parameter-Nutzung optional)	BOOL := FALSE	Fehlerliste zurücksetzen TRUE: Die Fehlerliste sowie die Anzahl der fehlerhaften Knoten am Ausgang ausgeben und anschließend zurücksetzen FALSE: Funktion wird nicht ausgeführt

Parameter der Ausgänge

7900

Parameter	Datentyp	Beschreibung
N_NODES	WORD	Anzahl der Knoten mit Heartbeat- oder Guarding-Fehlern 0 = kein Knoten hat einen Guarding- oder Heartbeat-Fehler
NODEID	ARRAY [0..MAXGUARDERROR] OF BYTE	Liste der Knoten-IDs mit Heartbeat- oder Guarding-Fehler. Der jüngste Eintrag steht im Index 0. MAXGUARDERROR ist abhängig vom Gerät → Kapitel ***Leistungsgrenzen der Geräte (CANopen)*** (→ Seite 38)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

CANOPEN_GETGUARDHBSTATSLV

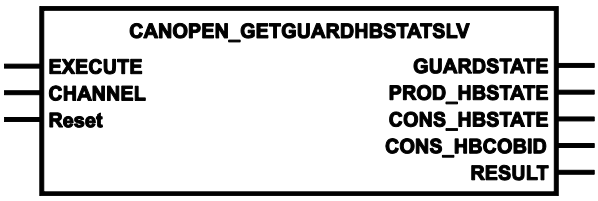
7902

= Get Guard and Heartbeat State Slave

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7904

CANOPEN_GETGUARDANDHBSTATESLAVE meldet der Steuerung im Slave-Betrieb folgende Zustände:
- Node-Guarding-Überwachung
- Heartbeat-Überwachung

Dabei kann die Steuerung Heartbeat-Producer und Heartbeat-Consumer sein.

Parameter der Eingänge

7905

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
Reset (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Die aktuellen Zustände an den Ausgängen ausgeben und anschließend auf "Kein Fehler" zurücksetzen FALSE: Funktion wird nicht ausgeführt

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7906

Parameter	Datentyp	Beschreibung
GUARDSTATE	BYTE	Status des Nodeguarding: 0 = 0x00 = kein Fehler (oder: inaktiv) 1 = 0x01 = Zeitüberschreitung (Konfiguration) 127 = 0x7F = noch keine Guarding-Nachricht empfangen
PROD_HBSTATE	BYTE	Steuerung als Heartbeat-Producer: 0 = 0x00 = inaktiv 1 = 0x01 = aktiv
CONS_HBSTATE	BYTE	Steuerung als Heartbeat-Consumer: 0 = 0x00 = kein Fehler 1 = 0x01 = Zeitüberschreitung (Konfiguration) 127 = 0x7F = noch keine Heartbeat-Nachricht empfangen
CONS_HBCOBID	WORD	COB-ID der Heartbeat-Nachricht, auf die der Consumer-Heartbeat der Steuerung hört (Konfiguration)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh

Bausteine: CANopen Emergency

Inhalt

CANOPEN_GETEMCYMESSAGES 133
CANOPEN_GETERRORREGISTER 135
CANOPEN_SENDEMCYMESSAGE 136

15073

© ifm electronic gmbh

CANOPEN_GETEMCYMESSAGES

7921

= Get Emergency-Messages

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:

Beschreibung

7923

CANOPEN_GETEMCYMESSAGES gibt alle Emergency-Nachrichten zurück, die die Steuerung seit dem letzten Löschen der Nachrichten von anderen Knoten am Netz empfangen hat.

Die Liste kann durch Setzen des entsprechenden Eingangs zurückgesetzt werden. Es werden maximal MAXEMCYMSGS Nachrichten gespeichert. Jede Nachricht enthält dabei als Info, von welchem Knoten sie gesendet wurde. Dabei steht die jüngste Nachricht im Index 0.

Parameter der Eingänge

7924

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
RstList (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Liste mit aufgelaufenen CAN-Nachrichten am Ausgang ausgeben und anschließend löschen FALSE: Funktion wird nicht ausgeführt

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7925

Parameter	Datentyp	Beschreibung
N_MSGS	DWORD	Anzahl der aufgelaufenen Nachrichten
EMCY	ARRAY [0..MAXEMCYMSGS] OF T_EMCY	Emergency-Nachrichten Der jüngste Eintrag steht im Index 0. Struktur von T_EMCY: .NODEID – ID des Knotens von dem die Nachricht kam .EEC – Emergency Error Code .ER – Error Register .MSEF – Manufacturer Specific Error Code MAXEMCYMSG = 10
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh

CANOPEN_GETERRORREGISTER

7915

= Get Error-Register

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:

Beschreibung

7917

CANOPEN_GETERRORREGISTER liest die Fehler-Register 0x1001 und 0x1003 der Steuerung aus.

Parameter der Eingänge

7918

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
Reset_1001 (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Fehler-Register 0x1001 zurücksetzen FALSE: Funktion wird nicht ausgeführt
Reset_1003 (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Fehler-Register 0x1003 zurücksetzen Anzahl der Einträge auf 0 setzen FALSE: Funktion wird nicht ausgeführt Die Einträge bleiben unverändert.

Parameter der Ausgänge

7919

Parameter	Datentyp	Beschreibung
ER	BYTE	Inhalt des Fehler-Registers 0x1001
ERROR_FIELD	ARRAY [0..MAXERR] OF DWORD	Inhalt des Error-Registers 0x1003 Index 0 = Anzahl der gespeicherten Fehler Index 1...MAXERR = gespeicherte Fehler Der jüngste Fehler steht im Index 1 voreingestellt: MAXERR = 5
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	FB ist aktiv – noch nicht bearbeitet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

CANOPEN_SENDEMCYMESSAGE

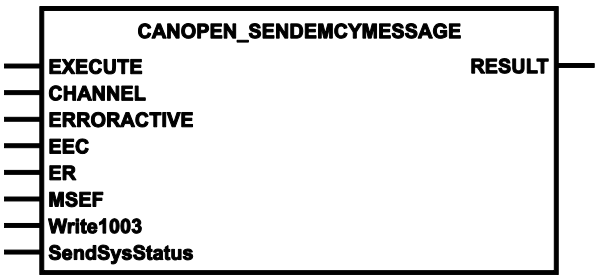
7908

= Send Emergency-Message

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CANopen_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7910

CANOPEN_SENDEMCYMESSAGE versendet eine EMCY-Nachricht. Die Nachricht wird aus den entsprechenden Parametern zusammengebaut und ins Register 0x1003 eingetragen. Die COB-ID für die Emergency-Nachricht wird aus den Konfigurationsdaten ermittelt.

Parameter der Eingänge

7911

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ERRORACTIVE	BOOL	FALSE ⇨ TRUE (Flanke): sendet den anstehenden Fehler-Code TRUE ⇨ FALSE (Flanke): Wenn der Fehler NICHT mehr ansteht, wird nach einer Verzögerung von ca. 1 s eine Null-Fehlermeldung gesendet.
EEC	WORD	EEC = Emergency Error Code = Fehlermeldungsnummer
ER (Parameter-Nutzung optional)	BYTE := 0	0 = Wert aus dem Fehler-Register 0x1001 verwenden
MSEF	ARRAY [0..4] OF BYTE	MSEF = Manufacturer Specific Error Code = Zusätzlicher Fehler-Code, der vom Hersteller festgelegt wird. Wert kommt aus der Anwendung.
Write1003 (Parameter-Nutzung optional)	BOOL := FALSE	TRUE: Diese EMCY-Nachricht im Objekt 0x1003 eintragen FALSE: Funktion wird nicht ausgeführt
SendSysStatus (Parameter-Nutzung optional)	BOOL := FALSE	Send System-Status TRUE: Der Systemstatus wird überprüft und bei Vorliegen eines Fehlerstatus wird dieser ins Netzwerk übertragen. FALSE: Funktion wird nicht ausgeführt

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7912

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	FB ist aktiv – noch nicht bearbeitet
39	27	kein Objekt 1001_{16} in der Konfiguration
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh
www.ifm.com

5.2.4 Bausteine: SAE J1939

Inhalt

Bausteine: SAE J1939 Status 138
Bausteine: SAE J1939 Request 146
Bausteine: SAE J1939 Empfangen 149
Bausteine: SAE J1939 Senden 154
Bausteine: SAE J1939 Diagnose 162

2273

Für SAE J1939 stellt **ifm electronic** eine Reihe von Bausteinen zur Verfügung, die im Folgenden erklärt werden.

Bausteine: SAE J1939 Status

Inhalt

J1939_ENABLE 139
J1939_GETDABYNAME 141
J1939_NAME 143
J1939_STATUS 145

15077

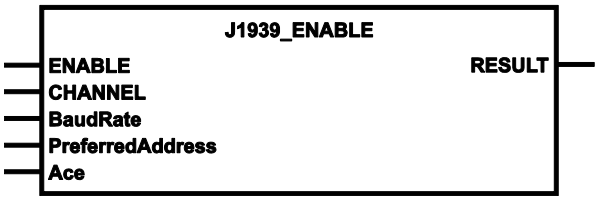
© ifm electronic gmbh

J1939_ENABLE

7641

Baustein-Typ = Funktionsbaustein (FB)
Baustein ist enthalten in Bibliothek `ifm_J1939_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7642

Zur Initialisierung des J1939-Stack wird J1939_ENABLE auf TRUE=1 gesetzt.

> Dieser FB startet auch die Soft-I/Os aus der CFG-Datei.

> Eine andere Baudrate wird nur übernommen, wenn CAN_ENABLE nicht bereits aufgerufen wurde.

ACE = **A**ddress **C**laiming **E**nable = Freigabe Adressanforderung:

- Wenn ein ifm-Controller via J1939 mit nur einem Motorsteuergerät kommuniziert:
  dann ACE = FALSE setzen.
- Wenn jedoch mehrere Motorsteuergeräte am selben Bus arbeiten:
  dann ACE = TRUE setzen.
  In diesem Fall müssen die Motorsteuergeräte das Address Claiming auch unterstützen!
  Andernfalls riskieren Sie Adress-Überschneidungen mit nachfolgendem Systemausfall.

Parameter der Eingänge

7643

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: J1939-Kanal freigeben Ace=TRUE: Adressanforderung erfolgt FALSE: J1939-Kanal sperren
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
BaudRate (Parameter-Nutzung optional)	WORD := 250	Baudrate [kBit/s] zulässige Werte: 20, 50, 100, 125, 250, 500, 800, 1 000
PreferredAddress (Parameter-Nutzung optional)	BYTE = 252	Bevorzugte Quell-Adresse
Ace (Parameter-Nutzung optional)	BOOL := TRUE	**A**ddress **C**laiming **E**nable = Freigabe Adressanforderung TRUE: Adressanforderung freigegeben (Steuergerät ist selbst-konfigurierend) FALSE: Keine Adressanforderung

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

8542

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist aktiv
9	09	CAN ist nicht aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

J1939_GETDABYNAME

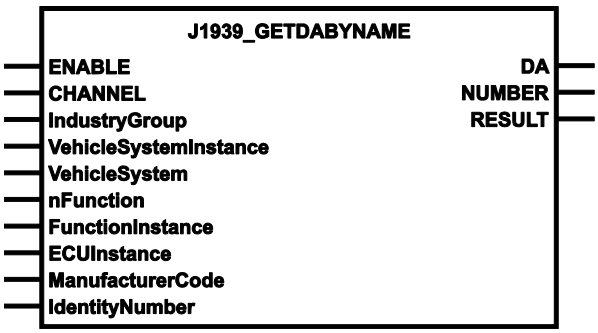
7664

= Get Destination Arbitrary Name

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_J1939_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7665

Über J1939_GETDABYNAME lässt sich anhand der Namensinformation die Ziel-Adresse eines oder mehrerer anderer Teilnehmer bestimmen.

- Wird an den optionalen Eingängen ein bestimmter Wert angelegt:
  ⇨ in der Ergebnisliste erscheinen nur die Teilnehmer, die diesen Wert besitzen.
- Wird an den optionalen Eingängen kein oder der voreingestellte Wert eingestellt:
  ⇨ bei der Filterung der Liste wird auf diesen Eintrag nicht geachtet.

© ifm electronic gmbh www.ifm.com

Parameter der Eingänge

7667

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
IndustryGroup (Parameter-Nutzung optional)	BYTE := 0xFF	Industry-Group = Industriegruppe des Geräts zulässige Werte = 0...7 255 = 0xFF = Filter für alle
VehicleSystemInstance (Parameter-Nutzung optional)	BYTE := 0xFF	Instanz des Fahrzeugsystems zulässige Werte = 0...15 = 0x00...0x0F 255 = 0xFF = Filter für alle
VehicleSystem (Parameter-Nutzung optional)	BYTE := 0xFF	Fahrzeugsystem zulässige Werte = 0...127 = 0x00...0x7F 255 = 0xFF = Filter für alle
nFunction (Parameter-Nutzung optional)	WORD := 0xFFFF	Funktionsnummer des Geräts zulässige Werte = 0...255 = 0x0000...0x00FF 65 535 = 0xFFFF = Filter für alle
FunctionInstance (Parameter-Nutzung optional)	BYTE := 0xFF	Instanz der Funktion zulässige Werte = 0...31 = 0x00...0x1F 255 = 0xFF = Filter für alle
ECUInstance (Parameter-Nutzung optional)	BYTE := 0xFF	Instanz des Steuergeräts zulässige Werte = 0...7 255 = 0xFF = Filter für alle
ManufacturerCode (Parameter-Nutzung optional)	WORD := 0xFFFF	Hersteller-Code (muss bei SAE beantragt werden) zulässige Werte = 0...2047 (2^{11}-1) = 0x0000...0x07FF 65 535 = 0xFFFF = Filter für alle
IdentityNumber (Parameter-Nutzung optional)	DWORD := 0xFFFF FFFF	Seriennummer des Geräts (sollte nicht überschrieben werden) zulässige Werte = 0...2047 (2^{11}-1) 4 294 967 295 = 0xFFFF FFFF = Filter für alle

Parameter der Ausgänge

7668

Parameter	Datentyp	Beschreibung
DA	ARRAY [0..254] OF BYTE	Liste mit den gefundenen Teilnehmern 255 = Teilnehmer mit dieser Nummer nicht gefunden
NUMBER	BYTE	Anzahl der gefundenen Busteilnehmer
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
8	08	Funktionsbaustein ist aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh

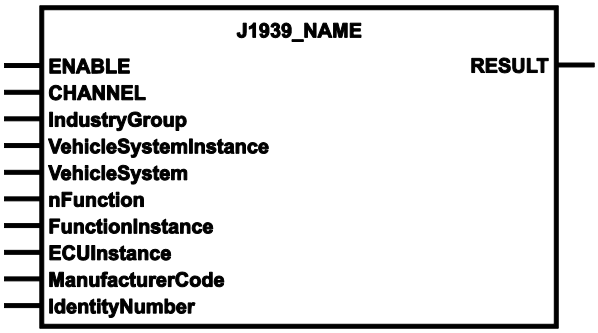
J1939_NAME

7646

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_J1939_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7648

Mit J1939_NAME kann dem Gerät ein Name gegeben werden, mit dem es sich im Netzwerk identifiziert.

Voreingestellt wird der Name der ifm verwendet.
Der Anwender hat die folgenden Möglichkeiten, den Namen des Gerätes zu ändern:

► die Informationen aus der CFG-Datei verwenden oder

► die gewünschten Daten mittels J1939_NAME überschreiben.

- Wird an den optionalen Eingängen kein oder der voreingestellte Wert eingestellt:
  ⇨ der voreingestellte Wert wird nicht überschrieben.

Die folgende Aufstellung zeigt die Zusammensetzung der 64-Bit-NAME-Information entsprechend SAE J1939-81:

Parameter	Datentyp	Beschreibung
Arbitrary Address Capable	1 Bit	beliebige Adresse verfügbar
Industry Group	3 Bit	Industriegruppe des Geräts
Vehicle System Instance	4 Bit	Instanz des Fahrzeugsystems
Vehicle System	7 Bit	Fahrzeugsystem
reserved	1 Bit	reserviert
Function	8 Bit	Funktion des Geräts
Function Instance	5 Bit	Instanz der Funktion
ECU Instance	3 Bit	Instanz der Steuerung
Manufacturer Code	11 Bit	Hersteller-Code (muss bei SAE beantragt werden)
Identify Number	21 Bit	Seriennummer des Geräts (sollte nicht überschrieben werden)

Tabelle: Zusammensetzung der 64-Bit-NAME-Information entsprechend SAE J1939-81

© ifm electronic gmbh www.ifm.com

Parameter der Eingänge

7652

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: beliebige Adresse verfügbar FALSE: feste Adresse
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
IndustryGroup (Parameter-Nutzung optional)	BYTE := 0xFF	Industry-Group = Industriegruppe des Geräts zulässige Werte = 0...7 255 = 0xFF = Filter für alle
VehicleSystemInstance (Parameter-Nutzung optional)	BYTE := 0xFF	Instanz des Fahrzeugsystems zulässige Werte = 0...15 = 0x00...0x0F 255 = 0xFF = Filter für alle
VehicleSystem (Parameter-Nutzung optional)	BYTE := 0xFF	Fahrzeugsystem zulässige Werte = 0...127 = 0x00...0x7F 255 = 0xFF = Filter für alle
nFunction (Parameter-Nutzung optional)	WORD := 0xFFFF	Funktionsnummer des Geräts zulässige Werte = 0...255 = 0x0000...0x00FF 65 535 = 0xFFFF = Filter für alle
FunctionInstance (Parameter-Nutzung optional)	BYTE := 0xFF	Instanz der Funktion zulässige Werte = 0...31 = 0x00...0x1F 255 = 0xFF = Filter für alle
ECUInstance (Parameter-Nutzung optional)	BYTE := 0xFF	Instanz des Steuergeräts zulässige Werte = 0...7 255 = 0xFF = Filter für alle
ManufacturerCode (Parameter-Nutzung optional)	WORD := 0xFFFF	Hersteller-Code (muss bei SAE beantragt werden) zulässige Werte = 0...2047 (2^{11}-1) = 0x0000...0x07FF 65 535 = 0xFFFF = Filter für alle
IdentityNumber (Parameter-Nutzung optional)	DWORD := 0xFFFF FFFF	Seriennummer des Geräts (sollte nicht überschrieben werden) zulässige Werte = 0...2047 (2^{11}-1) 4 294 967 295 = 0xFFFF FFFF = Filter für alle

Parameter der Ausgänge

7661

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist aktiv
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh

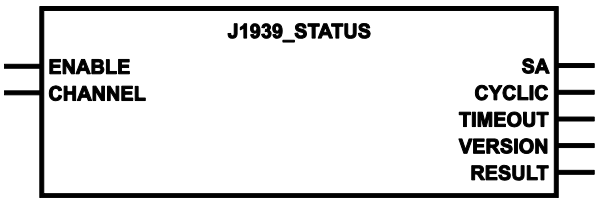
J1939_STATUS

7670

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_J1939_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7672

Mit J1939_STATUS können relevante Informationen zum J1939-Stack zurückgelesen werden.

Parameter der Eingänge

7673

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät

Parameter der Ausgänge

7674

Parameter	Datentyp	Beschreibung
SA	BYTE	aktuelle Quell-Adresse (z.B. nach Adress-Claiming)
CYCLIC	WORD	Anzahl der zyklischen Nachrichten
TIMEOUT	BYTE	Quell-Adresse des Knotens, der Daten für Prozessabbild nicht rechtzeitig zur Verfügung gestellt hat 255 = 0xFF = alle Knoten haben rechtzeitig gesendet
VERSION	DWORD	Version der ifm-CAN-Stack-Bibliothek
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Protokoll ist aktiv
2	02	Protokoll ist inaktiv
3	03	Source-Adresse angefordert
4	04	Adresse verloren
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

Bausteine: SAE J1939 Request

Inhalt

J1939_SPEC_REQ 147
J1939_SPEC_REQ_MULTI 148

15079

© ifm electronic gmbh www.ifm.com

J1939_SPEC_REQ

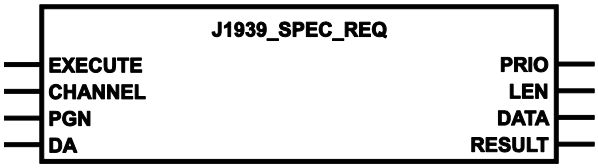
15023

= J1939 Specific Request

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_J1939_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

15026

J1939_SPECIFIC_REQUEST fragt eine spezifizierte Nachricht bei einer anderen Steuerung an und empfängt sie.

Beim Request einer Multiframe-Nachricht:
- der FB gibt die ersten 8 Bytes der Daten aus
- RESULT zeigt einen Fehler an

Parameter der Eingänge

15028

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
PGN	DWORD	PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer zulässig = 1...??? = 0x00000001...0x???
DA	BYTE	J1939-Adresse des angefragten Geräts

Parameter der Ausgänge

15029

Parameter	Datentyp	Beschreibung
PRIO	BYTE	Nachrichten-Prioritätin der PDU (Parameter Data Unit) zulässig = 0…7
LEN	WORD	Anzahl der empfangenen Bytes (0...8)
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist aktiv – noch keine Daten empfangen
64	40	Fehler: Multiframe empfangen
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

J1939_SPEC_REQ_MULTI

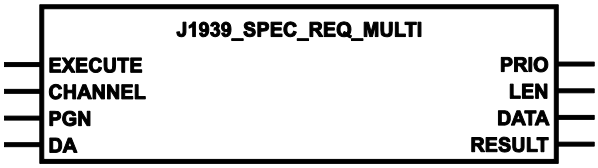
15033

= J1939 Specific Request Multiframe Message

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_J1939_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

15036

J1939_SPECIFIC_REQUEST fragt eine spezifizierte Multiframe-Nachricht bei einer anderen Steuerung an und empfängt sie.

Parameter der Eingänge

15037

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
PGN	DWORD	PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer zulässig = 1...??? = 0x00000001...0x???
DA	BYTE	J1939-Adresse des angefragten Geräts

Parameter der Ausgänge

15038

Parameter	Datentyp	Beschreibung
PRIO	BYTE	Nachrichten-Prioritätin der PDU (Parameter Data Unit) zulässig = 0…7
LEN	WORD	Anzahl der zu übertragenden Daten-Bytes zulässig = 1...1 785 = 0x0001...0x06F9
DATA	ARRAY [0..1784] OF BYTE	Empfangene Daten (1...1785 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist aktiv – noch keine Daten empfangen
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

Bausteine: SAE J1939 Empfangen

Inhalt

J1939_RX 150
J1939_RX_FIFO 151
J1939_RX_MULTI 153

15081

© ifm electronic gmbh www.ifm.com

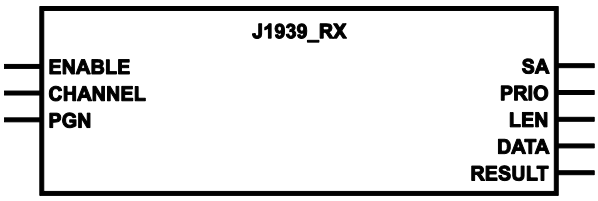
J1939_RX

7724

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_J1939_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7725

J1939_RX ist die einfachste Methode zum Empfangen von Single-Frame-Nachrichten. Es wird die zuletzt auf dem CAN-Bus gelesene Nachricht zurückgegeben.

Parameter der Eingänge

7726

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
PGN	DWORD	PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer zulässig = 0...262 143 = 0x00000000...0x0003FFFF

Die PGN = 0 wird nicht verwendet.

Parameter der Ausgänge

7727

Parameter	Datentyp	Beschreibung
SA	BYTE	**S**ource **A**ddress des Senders
PRIO	BYTE	Nachrichten-Prioritätin der PDU (Parameter Data Unit) zulässig = 0…7
LEN	WORD	Anzahl der empfangenen Bytes (0...8)
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist noch nicht ausgeführt
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

J1939_RX_FIFO

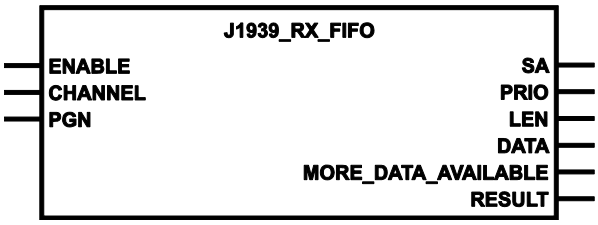
7732

= J1939 RX with FIFO

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_J1939_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7733

J1939_RX_FIFO ermöglicht es, alle spezifizierten Nachrichten zu empfangen und nacheinander aus einem FIFO zu lesen.

Parameter der Eingänge

7734

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
PGN	DWORD	PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer zulässig = 0...262 143 = 0x00000000...0x0003FFFF

Die PGN = 0 wird nicht verwendet.

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7735

Parameter	Datentyp	Beschreibung
SA	BYTE	**S**ource **A**ddress des Senders
PRIO	BYTE	Nachrichten-Prioritätin der PDU (Parameter Data Unit) zulässig = 0…7
LEN	BYTE	Anzahl der empfangenen Bytes (0...8)
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
MORE_DATA_AVAILABLE	BOOL	TRUE: weitere empfangene Daten im FiFo vorhanden FALSE: keine weiteren Daten im FiFo vorhanden
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist aktiv – noch keine Daten empfangen
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren

© ifm electronic gmbh www.ifm.com

J1939_RX_MULTI

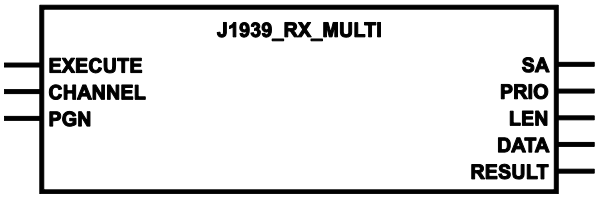
7736

= J1939 RX Multiframe Message

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_J1939_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7741

Mit J1939_RX_MULTI ist der Empfang von Multiframe-Nachrichten möglich.

Parameter der Eingänge

7743

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
PGN	DWORD	PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer zulässig = 0...262 143 = 0x00000000...0x0003FFFF

Die PGN = 0 wird nicht verwendet.

Parameter der Ausgänge

7744

Parameter	Datentyp	Beschreibung
SA	BYTE	**S**ource **A**ddress des Senders
PRIO	BYTE	Nachrichten-Prioritätin der PDU (Parameter Data Unit) zulässig = 0…7
LEN	WORD	Anzahl der empfangenen Bytes zulässige Werte = 0000 0000...0000 $06F9_{16}$ = 0...1 785_{10}
DATA	ARRAY [0..1784] OF BYTE	Empfangene Daten (1...1785 Bytes)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
5	05	FB ist aktiv – noch keine Daten empfangen
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

Bausteine: SAE J1939 Senden

Inhalt

J1939_TX 155
J1939_TX_ENH 156
J1939_TX_ENH_CYCLIC 158
J1939_TX_ENH_MULTI 160

15083

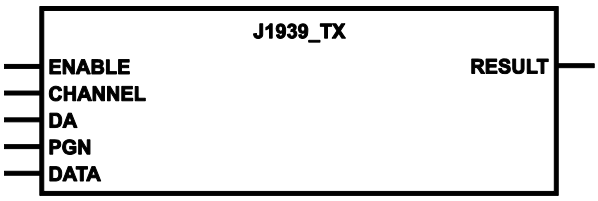
© ifm electronic gmbh

J1939_TX

7688

Baustein-Typ = Funktionsbaustein (FB)
Baustein ist enthalten in Bibliothek `ifm_J1939_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7689

J1939_TX ist die einfachste Methode zum Versenden von Single-Frame-Nachrichten.

Parameter der Eingänge

7690

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DA	BYTE := 249	DA = **D**estination **A**ddress = Zieladresse der ECU PGN > 61139: Parameter DA wird ignoriert
PGN	DWORD	PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer zulässig = 0...262 143 = 0x00000000...0x0003FFFF
DATA	ARRAY [0..7] OF BYTE	zu sendende Daten (1...8 Bytes)

Parameter der Ausgänge

7693

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren

© ifm electronic gmbh www.ifm.com

J1939_TX_ENH

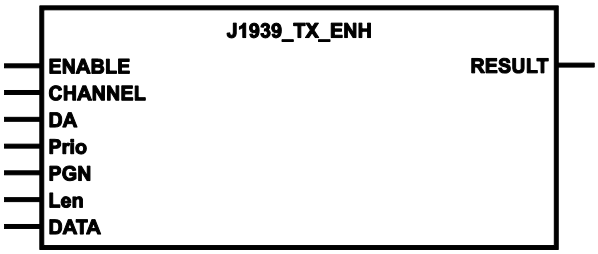
7696

= J1939 TX enhanced

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_J1939_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7697

Zusätzliche Einstellmöglichkeiten bietet J1939_TX_ENH (für: enhanced) für Single-Frame-Nachrichten:

- Sende-Priorität
- Datenlänge

Multi-Frame Nachrichten → ***J1939_TX_ENH_MULTI*** (→ Seite 160).

Parameter der Eingänge

7702

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DA	BYTE := 249	DA = **D**estination **A**ddress = Zieladresse der ECU PGN > 61139: Parameter DA wird ignoriert
Prio (Parameter-Nutzung optional)	BYTE := 3	Nachrichten-Priorität zulässige Werte = 0…7
PGN	DWORD	PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer zulässig = 0...262 143 = 0x00000000...0x0003FFFF
Len (Parameter-Nutzung optional)	BYTE := 8	Anzahl der zu sendenden Bytes zulässige Werte = 0...8
DATA	ARRAY [0..7] OF BYTE	zu sendende Daten (1...8 Bytes)

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7969

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
242	F2	Fehler: Einstellung ist nicht möglich
250	FA	Fehler: FiFo ist voll – Daten wurden verloren

© ifm electronic gmbh www.ifm.com

J1939_TX_ENH_CYCLIC

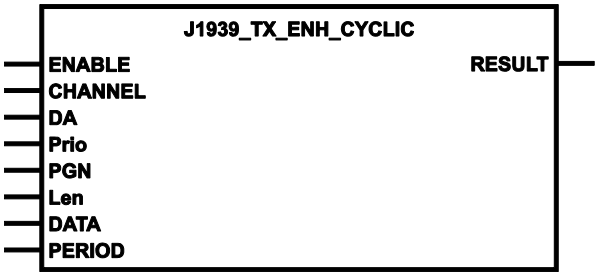
7716

= J1939 TX enhanced Cyclic

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_J1939_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7718

J1939_TX_ENH_CYCLIC dient dem zyklischen Versand von CAN-Nachrichten.

Der FB entspricht ansonsten ***J1939_TX_ENH*** (→ Seite 156).

► Mit dem Parameter PERIOD die Periodendauer einstellen.

Eine zu kurze Periodendauer kann zu einer hohen Buslast führen!
Die Buslast kann das Verhalten des Gesamtsystems beinträchtigen.

Parameter der Eingänge

7719

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DA	BYTE := 249	DA = **D**estination **A**ddress = Zieladresse der ECU PGN > 61139: Parameter DA wird ignoriert
Prio (Parameter-Nutzung optional)	BYTE := 3	Nachrichten-Priorität zulässige Werte = 0…7
PGN	DWORD	PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer zulässig = 0...262 143 = 0x00000000...0x0003FFFF
Len (Parameter-Nutzung optional)	BYTE := 8	Anzahl der zu sendenden Bytes zulässige Werte = 0...8
DATA	ARRAY [0..7] OF BYTE	zu sendende Daten (1...8 Bytes)
PERIOD	TIME	Periodendauer

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7720

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
8	08	Funktionsbaustein ist noch nicht ausgeführt
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

J1939_TX_ENH_MULTI

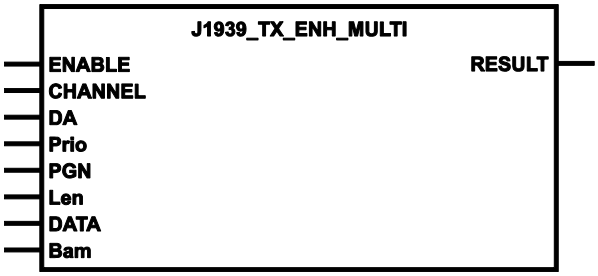
7699

= J1939 TX enhanced Multiframe Message

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_J1939_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7705

Die Übertragung von Multi-Frame-Nachrichten erfolgt mit J1939_TX_ENH_MULTI.

Der FB entspricht ***J1939_TX_ENH*** (→ Seite 156). Zusätzlich kann hier bestimmt werden, ob die Übertragung als BAM (**B**roadcast **A**nnounce **M**essage) erfolgen soll.

Parameter der Eingänge

7712

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DA	BYTE := 249	DA = **D**estination **A**ddress = Zieladresse der ECU PGN > 61139: Parameter DA wird ignoriert
Prio (Parameter-Nutzung optional)	BYTE := 3	Nachrichten-Priorität zulässige Werte = 0…7
PGN	DWORD	PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer zulässig = 0...262 143 = 0x00000000...0x0003FFFF
Len (Parameter-Nutzung optional)	WORD := 8	Anzahl der zu übertragenden Daten-Bytes zulässig = 1...1 785 = 0x0001...0x06F9
DATA	ARRAY [0..1784] OF BYTE	Zu sendende Daten (1...1785 Bytes)
Bam (Parameter-Nutzung optional)	BOOL := FALSE	BAM = **B**roadcast **A**nnounce **M**essage = Nachricht an alle Teilnehmer TRUE: Multi-Frame-Übertragung als BAM Nachricht an alle Teilnehmer FALSE: automatisch; Nachricht nur an Zieladresse

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7714

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
8	08	Funktionsbaustein ist noch nicht ausgeführt
65	41	Fehler: senden ist nicht möglich
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

Bausteine: SAE J1939 Diagnose

Inhalt

J1939_DM1RX 163
J1939_DM1TX 165
J1939_DM1TX_CFG 168
J1939_DM3TX 169

15085

© ifm electronic gmbh www.ifm.com

J1939_DM1RX

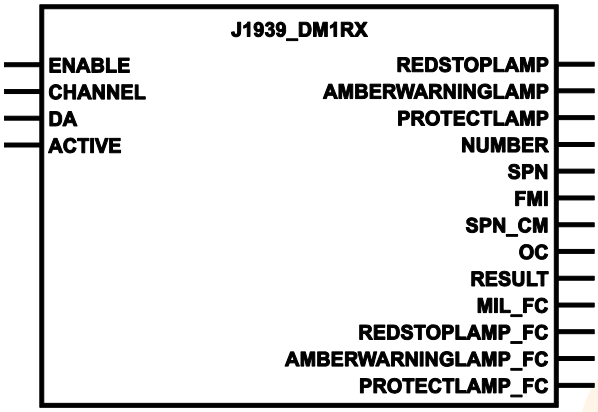
14977

= J1939 Diagnostic Message 1 RX

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_J1939_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7761

J1939_RX_DM1 empfängt Diagnosemeldungen DM1 oder DM2 von anderen ECUs.

Parameter der Eingänge

14979

Parameter	Datentyp	Beschreibung
ENABLE	BOOL := FALSE	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DA	BYTE	DA = **D**estination **A**ddress = Zieladresse der ECU, von der die DTCs geholt werden sollen. DA = 254: DTCs aus Gerät selbst lesen
ACTIVE	BOOL	TRUE: aktive DTCs (DM1) lesen FALSE: davor aktive DTCs (DM2) lesen

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

14980

Parameter	Datentyp	Beschreibung
REDSTOPLAMP	BOOL	Rote Stopp-Lampe (nur für ältere Projekte) TRUE: EIN FALSE: AUS
AMBERWARNINGLAMP	BOOL	Gelbe Warn-Lampe (nur für ältere Projekte) TRUE: EIN FALSE: AUS
PROTECTLAMP	BOOL	Schutz-Lampe (nur für ältere Projekte) TRUE: EIN FALSE: AUS
NUMBER	BYTE	Anzahl der empfangenen DTCs (0...8)
SPN	WORD	**S**uspect **P**arameter **N**umber
FMI	BYTE	**F**ailure-**M**ode-**I**ndicator zulässige Werte = 0...31 = 0x00...0x1F
SPN_CM	BOOL	Conversion Method
OC	BYTE	Occurrence Count = Ereigniszähler
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)
MIL_FC	BYTE	Status der elektronischen Komponente: Fehlfunktion-Anzeigelampe Status und Blink-Code: 0 = Aus 1 = Ein 2 = langsam blinken 3 = schnell blinken
REDSTOPLAMP_FC	BYTE	Status der elektronischen Komponente: Rote Stopp-Lampe Status und Blink-Code: 0 = Aus 1 = Ein 2 = langsam blinken 3 = schnell blinken
AMBERWARNINGLAMP_FC	BYTE	Status der elektronischen Komponente: Gelbe Warnlampe Status und Blink-Code: 0 = Aus 1 = Ein 2 = langsam blinken 3 = schnell blinken
PROTECTLAMP_FC	BYTE	Status der elektronischen Komponente: Schutz-Lampe Status und Blink-Code: 0 = Aus 1 = Ein 2 = langsam blinken 3 = schnell blinken

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
8	08	FB ist aktiv – keine Daten wurden empfangen
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

J1939_DM1TX

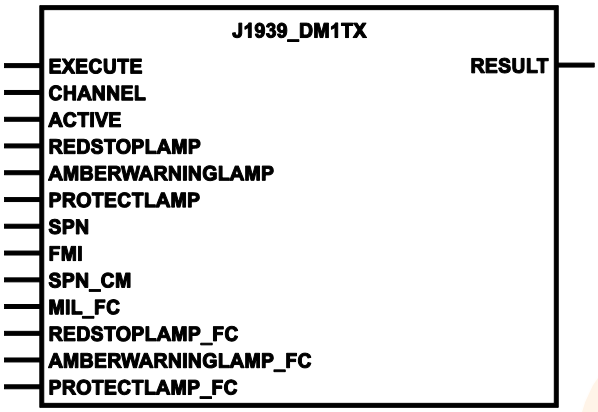
14993

= J1939 Diagnostic Message 1 TX

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_J1939_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

7747

Mit J1939_TX_DM1 (DM = **D**iagnostic-**M**essage) kann die Steuerung nur eine aktive Fehlermeldung an den CAN-Stack übergeben.

> Diese Meldung wird in die Hardware-Konfiguration gesichert

> Meldung wird als aktiv markiert und sekündlich als DM1 gesendet.

> Falls der Fehler bereits auftrat, wird der Ereignis-Zähler inkrementiert.
  Der Ereignis-Zähler wird vom CAN-Stack verwaltet.

> Es erfolgt eine ODER-Verknüpfung aller Bits der Trouble-Codes. Sobald in einem der Trouble-Codes ein Bit gesetzt ist, ist es auch im Lampenstatus gesetzt.

Sobald eine Anfrage nach DM2 kommt, kann der CAN-Stack die entsprechenden Informationen aus der Hardware-Konfiguration auslesen und versenden.

> Bei Eintreffen einer DM3-Nachricht werden alle nicht aktiven Fehler im Fehlerspeicher in der Hardware-Konfiguration gelöscht.

© ifm electronic gmbh www.ifm.com

Parameter der Eingänge

14995

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
ACTIVE	BOOL	TRUE: DTC ist aktiv Wird zyklisch (1x je Sekunde) als DM1 gesendet FALSE: DTC ist nicht mehr aktiv Wird in der Hardware-Konfiguration gesichert Wird bei Anfrage als DM2 gesendet
REDSTOPLAMP	BOOL	Rote Stopp-Lampe (nur für ältere Projekte) TRUE: EIN FALSE: AUS
AMBERWARNINGLAMP	BOOL	Gelbe Warn-Lampe (nur für ältere Projekte) TRUE: EIN FALSE: AUS
PROTECTLAMP	BOOL	Schutz-Lampe (nur für ältere Projekte) TRUE: EIN FALSE: AUS
SPN	WORD	**S**uspect **P**arameter **N**umber
FMI	BYTE	**F**ailure-**M**ode-**I**ndicator zulässige Werte = 0...31 = 0x00...0x1F
SPN_CM	BOOL	Conversion Method
MIL_FC	BYTE	Status der elektronischen Komponente: Fehlfunktion-Anzeigelampe Status und Blink-Code: 0 = Aus 1 = Ein 2 = langsam blinken 3 = schnell blinken
REDSTOPLAMP_FC	BYTE	Status der elektronischen Komponente: Rote Stopp-Lampe Status und Blink-Code: 0 = Aus 1 = Ein 2 = langsam blinken 3 = schnell blinken
AMBERWARNINGLAMP_FC	BYTE	Status der elektronischen Komponente: Gelbe Warnlampe Status und Blink-Code: 0 = Aus 1 = Ein 2 = langsam blinken 3 = schnell blinken
PROTECTLAMP_FC	BYTE	Status der elektronischen Komponente: Schutz-Lampe Status und Blink-Code: 0 = Aus 1 = Ein 2 = langsam blinken 3 = schnell blinken

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

7750

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Daten wurden in Fehlerspeicher aktiv gekennzeichnet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

J1939_DM1TX_CFG

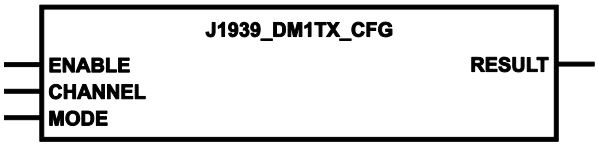
15424

= J1939 Diagnostic Message 1 TX configurable

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_J1939_NT_V02.00.02.LIB` oder höher

Symbol in CODESYS:



Beschreibung

15426

Ab Laufzeitsystem V03.00.03 sendet der CAN-Stack automatisch sekündliche DM1-Nachrichten, sobald der FB ***J1939_ENABLE*** (→ Seite 139) für das betreffende CAN-Interface aufgerufen wurde.

► Den FB J1939_DM1TX_CFG nutzen, wenn dieses automatische, zyklische Senden von DM1-Nachrichten des CAN-Stacks nicht gewünscht ist.

Folgende Modi für die zyklische Sendung von DM1-Nachrichten stehen mit dem FB zur Verfügung:

MODE = 0 (voreingestellt)	Der CAN-Stack sendet normkonform, sekündlich DM1 "zero active faults"-Nachrichten. Manuelles Senden von DM1-Nachrichten über den FB ***J1939_DM1TX*** (→ Seite 165) ist möglich.
MODE = 1	Der CAN-Stack sendet keine zyklischen DM1 "zero active faults"-Nachrichten. Auf DM2-Anfragen wird automatisch geantwortet. manuelles Senden von DM1-Nachrichten über den FB ***J1939_DM1TX*** (→ Seite 165) ist möglich.
MODE = 2	Der CAN-Stack sendet keine zyklischen DM1 "zero active faults"-Nachrichten Der CAN-Stack sendet auch keine automatische Antwort auf DM2-Anfragen.

Parameter der Eingänge

15427

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
MODE	BYTE := 0	Betriebsart des Bausteins zulässig = 0...2 (→ Beschreibung des FBs)

Parameter der Ausgänge

15429

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

J1939_DM3TX

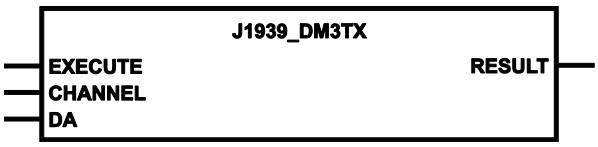
15002

= J1939 Diagnostic Message 3 TX

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_J1939_NT_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

15004

J1939_DM3TX (DM = **D**iagnostic-**M**essage) ermöglicht das Löschen der inaktiven DTCs auf einem anderen Gerät.

> Bei Eintreffen einer DM3-Nachricht werden alle nicht aktiven Fehler im Fehlerspeicher in der Hardware-Konfiguration gelöscht.

Parameter der Eingänge

15006

Parameter	Datentyp	Beschreibung
EXECUTE	BOOL := FALSE	FALSE ⇨ TRUE (Flanke): Baustein einmalig ausführen sonst: Baustein nicht aktiv ein bereits gestarteter Baustein wird abgearbeitet
CHANNEL	BYTE	CAN-Schnittstelle (1...n) je nach Gerät
DA	BYTE	DA = **D**estination **A**ddress = Zieladresse der ECU, auf der die DTCs gelöscht werden sollen. DA = 254: DTCs (DM2) im Gerät selbst löschen

Parameter der Ausgänge

15008

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
242	F2	Fehler: Einstellung ist nicht möglich

© ifm electronic gmbh www.ifm.com

5.2.5 Bausteine: Eingangswerte verarbeiten

Inhalt

FASTCOUNT 171
INC_ENCODER 173
INPUT 176
PERIOD 178

1302

Hier zeigen wir Ihnen ifm-Funktionsbausteine zum Lesen und Verarbeiten der analogen oder binären Signale am Geräte-Eingang.

© ifm electronic gmbh www.ifm.com

FASTCOUNT

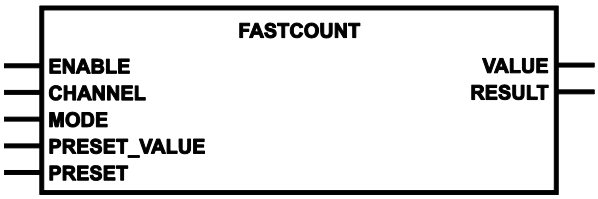
8112

= Fast Count

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

8114

FASTCOUNT arbeitet als Zählerbaustein für schnelle Eingangsimpulse (bis 30 kHz).
Dieser FB erfasst Impulse an den schnellen Eingangskanälen (→ Datenblatt).

Überlauf oder Unterlauf des Zählerwerts wird nicht erkannt.

Parameter der Eingänge

8115

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > gestartete Prozesse laufen im Hintergrund weiter > FB-Ausgänge werden nicht aktualisiert
CHANNEL	BYTE	Nummer des schnellen Eingangskanals (0...3) 0...3 für die Eingänge I00...I03
MODE	BYTE	Betriebsart des Bausteins: 0 = 0x00 = Zähler stoppen 21 = 0x15 = Aufwärts-Zähler 22 = 0x16 = Abwärts-Zähler
PRESET_VALUE	DWORD	Zähler-Startwert
PRESET	BOOL	TRUE (nur 1 Zyklus lang): Zähler-Startwert PRESET_VALUE laden FALSE: Zähler ist aktiv

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

8116

Parameter	Datentyp	Beschreibung
VALUE	DWORD	Ausgabewert
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
2	02	Funktionsbaustein ist aktiv (Aktion noch nicht beendet)
3	03	Funktionsbaustein ist aktiv – es liegen noch keine gültigen Werte vor
130	82	Kanaleinstellung ist ungültig
132	84	Moduseinstellung ist ungültig

© ifm electronic gmbh www.ifm.com

INC_ENCODER

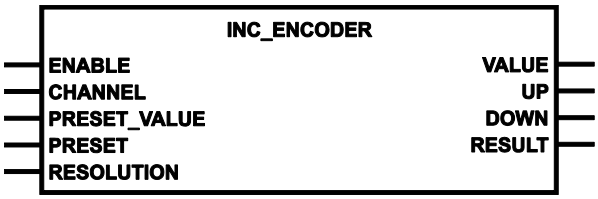
8134

= Incremental Encoder

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

8135

INC_ENCODER organisiert Vorwärts-/Rückwärts-Zählerfunktion zur Auswertung von Drehgebern.

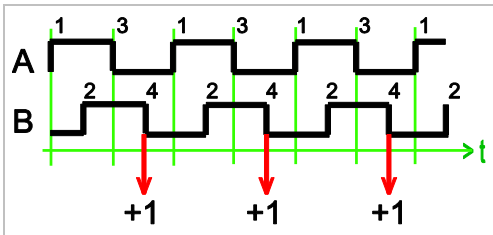
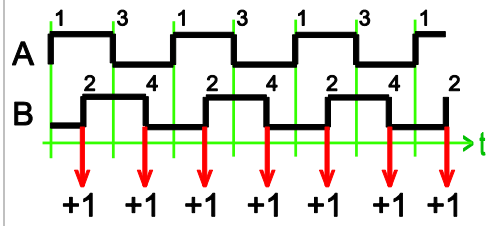
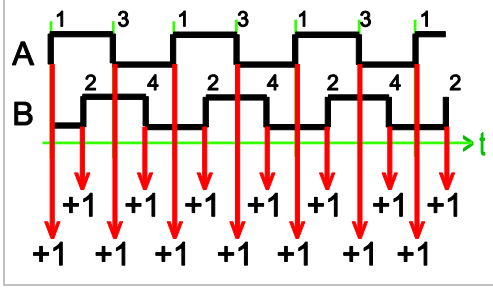
Immer zwei Frequenzeingänge bilden das Eingangspaar, das mit dem FB ausgewertet wird.
Zulässige Eingangsfrequenz = 0...1 000 Hz

Über den PRESET_VALUE kann der Zähler auf einen Voreinstellwert gesetzt werden. Der Wert wird übernommen, wenn PRESET auf TRUE gesetzt wird. Anschließend muss PRESET wieder auf FALSE gesetzt werden, damit der Zähler wieder aktiv wird.

Am Ausgang VALUE steht der aktuelle Zählerstand an. Die Ausgänge UP und DOWN zeigen die letzte Zählrichtung des Zählers an. Die Ausgänge sind dann TRUE, wenn der Zähler in die entsprechende Richtung gezählt hat. Wurde der Drehgeber seit dem letzten Aufruf des Bausteins nicht verändert, sind beide Ausgänge FALSE.

Am Eingang RESOLUTION kann die Auflösung des Drehgebers vervielfacht ausgewertet werden:
1 = normale Auflösung (-536 870 912...536 870 911, identisch mit der Auflösung des Drehgebers),
2 = Auflösung doppelt auswerten (-1 073 741 824...1 073 741 823),
4 = Auflösung 4-fach auswerten (-2 147 483 648...2 147 483 647).
Alle anderen Werte an diesem Eingang bedeuten normale Auflösung.

© ifm electronic gmbh www.ifm.com

	RESOLUTION = 1 zählt bei jeder vierten Flanke (= Auflösung des Drehgebers)
	RESOLUTION = 2 zählt bei jeder zweiten Flanke
	RESOLUTION = 4 zählt bei jeder steigenden und fallenden Flanke

Parameter der Eingänge

8137

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > gestartete Prozesse laufen im Hintergrund weiter > FB-Ausgänge werden nicht aktualisiert
CHANNEL	BYTE	Nummer des Eingangskanal-Paares (0/2): 0 = Kanalpaar 0 = Eingänge I00 + I01 2 = Kanalpaar 1 = Eingänge I02 + I03
PRESET_VALUE	DINT	Zähler-Startwert
PRESET	BOOL	TRUE (nur 1 Zyklus lang): Zähler-Startwert PRESET_VALUE laden FALSE: Zähler ist aktiv
RESOLUTION	BYTE	Auswertung der Drehgeber-Auflösung: 01 = zählt bei jeder vierten Flanke (= Auflösung des Drehgebers) 02 = zählt bei jeder zweiten Flanke 04 = zählt bei jeder steigenden und fallenden Flanke Alle anderen Werte zählen wie "01".

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

8138

Parameter	Datentyp	Beschreibung
VALUE	DINT	wenn RESOLUTION = 1: VALUE = –536 870 912...536 870 911 (= ¼ Bereich von DINT) wenn RESOLUTION = 2: VALUE = –1 073 741 824...1 073 741 823 (= ½ Bereich von DINT) wenn RESOLUTION = 4: VALUE = –2 147 483 648…2 147 483 647 (= Bereich von DINT)
UP	BOOL	TRUE: Zähler zählte im letzten Zyklus aufwärts FALSE: Zähler zählte im letzten Zyklus nicht aufwärts
DOWN	BOOL	TRUE: Zähler zählte im letzten Zyklus abwärts FALSE: Zähler zählte im letzten Zyklus nicht abwärts
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
2	02	Funktionsbaustein ist aktiv (Aktion noch nicht beendet)
3	03	Funktionsbaustein ist aktiv – es liegen noch keine gültigen Werte vor
130	82	Kanaleinstellung ist ungültig
138	8A	Auflösungseinstellung ist ungültig

© ifm electronic gmbh www.ifm.com

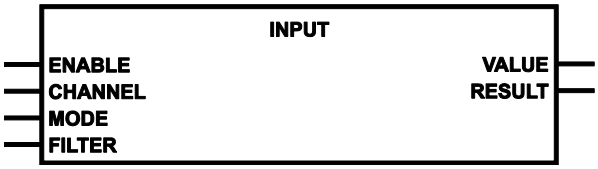
INPUT

8103

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

16650

INPUT weist einem Eingangskanal eine Betriebsart zu (→ Datenblatt). Der FB ermöglicht die Zustandserfassung am gewählten Kanal.

Die Messung und der Ausgangswert resultieren aus der über MODE angegebenen Betriebsart:
- binärer Eingang plus-schaltend (BL) für positives Gebersignal (mit/ohne Diagnose)
- binärer Eingang, minus-schaltend (BH) für negatives Gebersignal
- analoger Eingang 0...20 mA
- analoger Eingang 0...10 V
- analoger Eingang 0...32 V
- analoger Eingang ratiometrisch 0...32 V
- analoger Eingang Widerstandsmessung 16...30 000 Ω

Im laufenden Betrieb sollte die Betriebsart nicht geändert werden.

Die Analogwerte werden normiert ausgegeben.

© ifm electronic gmbH www.ifm.com

Parameter der Eingänge

16653

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	Nummer des Eingangskanals (0...7) 0...7 für die Eingänge IN0...IN7
MODE	BYTE	Betriebsart des Eingangskanals:

0 = 0x00	Aus	
1 = 0x01	(nur für binär ausgewertete Eingänge) Binäreingang, plus-schaltend (BL)	
3 = 0x03	Spannungseingang	0…10 000 mV
6 = 0x06	Spannungseingang, ratiometrisch	0...1 000 ‰
7 = 0x07	Stromeingang	0…20 000 µA
9 = 0x09	Spannungseingang	0…32 000 mV
10 = 0x0A	(nur für analog ausgewertete Eingänge) Binäreingang, plus-schaltend (BL)	
11 = 0x0B	(nur für analog ausgewertete Eingänge) Binäreingang, plus-schaltend (BL) mit Diagnose (Namur)	
12 = 0x0C	Binäreingang, minus-schaltend (BH)	
18 = 0x12	Widerstandseingang	16…30 000 Ω

FILTER	BYTE	Filter für die Messung am Eingang: zulässig = 0...8 empfohlen = 4 → Kapitel ***Software-Filter der Eingänge konfigurieren*** (→ Seite 49)

Parameter der Ausgänge

8106

Parameter	Datentyp	Beschreibung
VALUE	WORD	aktueller Wert oder Zustand des Eingangskanals
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
2	02	Funktionsbaustein ist aktiv (Aktion noch nicht beendet)
3	03	Funktionsbaustein ist aktiv – es liegen noch keine gültigen Werte vor
130	82	Kanaleinstellung ist ungültig
132	84	Moduseinstellung ist ungültig
136	88	Filtereinstellung ist ungültig
141	8D	Leiterbruch ist aufgetreten
142	8E	Schluss gegen Versorgung ist aufgetreten
144	90	Strom am Eingang ist zu hoch

© ifm electronic gmbh www.ifm.com

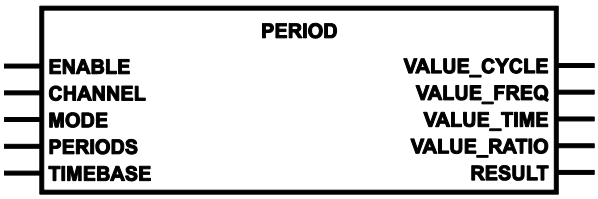
PERIOD

8122

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

15850

PERIOD misst die Frequenz in [Hz] oder die Periodendauer (Zykluszeit) in [µs] oder die Phasenverschiebung in [°] am angegebenen Kanal, je nach eingestellter Betriebsart:

MODE dez	MODE hex	Beschreibung
0	00	keine Messung
14	0E	Frequenzmessung Die positiven Flanken über eine bestimmte Zeit zählen.
19	13	Periodendauermessung (besser ersetzen durch MODE = 20!) Den Zeitabstand zwischen zwei positiven Flanken messen. Den Mittelwert über eine bestimmte Anzahl von Perioden angeben.
20	14	Periodendauer- und Ratiomessung Den Zeitabstand zwischen zwei positiven Flanken messen. Den Mittelwert über eine bestimmte Anzahl von Perioden angeben.
25	19	(ab LZS-Version 03.02.zz) Phasenverschiebung (0...359°) zwischen Kanal A und Kanal B eines Eingangssignalpaars (Mittelung nur sinnvoll, wenn keine großen Sprünge > 179° im System auftreten können)

Im laufenden Betrieb sollte die Betriebsart nicht geändert werden.

Falls MODE=19 oder MODE=20 oder MODE=25:
Zulässige Eingangsfrequenz = 0,1...3 000 Hz.
Bei zu hoher Belastung kann die Zykluszeit unzulässig lang werden.
→ Kapitel *Leistungsgrenzen des Geräts* (→ Seite 37)

Parameter der Eingänge

8124

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > gestartete Prozesse laufen im Hintergrund weiter > FB-Ausgänge werden nicht aktualisiert
CHANNEL	BYTE	(MODE = 14 / 19 / 20) Nummer des schnellen Eingangskanals (0...3) 0...3 für die Eingänge IN0...IN3 (MODE = 25) Nummer des schnellen Eingangs-A-Kanals (0 / 2) 0 / 2 für die Eingänge IN0 / IN2 B-Kanal = A-Kanal + 1

© ifm electronic gmbh www.ifm.com

Parameter	Datentyp	Beschreibung
MODE	BYTE	Betriebsart des Bausteins: 0 = 0x00 = keine Messung 14 = 0x0E = Frequenzmessung 19 = 0x13 = Periodendauermessung 20 = 0x14 = Periodendauer- und Ratiomessung 25 = 0x19 = Phasenverschiebung zweier Eingangssignale
PERIODS	BYTE	Anzahl der Perioden, über die gemittelt wird (1...4) • wenn MODE = 14 / 19 / 20 ⇨ arithmetisch mitteln • wenn MODE = 25 ⇨ geometrisch mitteln • wenn PERIODS = 1 ⇨ keine Mittelung
TIMEBASE	TIME	(nur relevant bei MODE = 14) Zeitdauer zum Zählen der Flanken in [ms] zulässige Werte = 1...2 000

Parameter der Ausgänge

8125

Parameter	Datentyp	Beschreibung
VALUE_CYCLE	DWORD	(MODE = 14 / 19 / 20) Zykluszeit in [µs] am Eingang (MODE = 25) Zykluszeit in [µs] am Kanal A des Eingangspaars
VALUE_FREQ	REAL	(MODE = 14 / 19 / 20) Frequenz der erfassten Perioden in [Hz] am Eingang (MODE = 25) Frequenz der erfassten Perioden in [Hz] am Kanal A des Eingangspaars
VALUE_TIME	TIME	(MODE = 14) Wert = 0 (nicht erfasst) (MODE = 19 / 20) Verstrichene Zeit seit der letzten positiven Flanke (MODE = 25) Verstrichene Zeit seit der letzten gültigen Messung
VALUE_RATIO	WORD	(MODE = 14) Wert = 0 (nicht erfasst) (MODE = 19 / 20) Puls-/Pause-Verhältnis des Eingangssignals in [‰] (MODE = 25) Phasenverschiebung in [°]
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
2	02	Funktionsbaustein ist aktiv (Aktion noch nicht beendet)
3	03	Funktionsbaustein ist aktiv – es liegen noch keine gültigen Werte vor
130	82	Kanaleinstellung ist ungültig
132	84	Moduseinstellung ist ungültig
137	89	Wert für PERIODS oder TIMEBASE ist ungültig
146	92	Periodendauer zu groß Bei MODE=25 auch: fehlende Flanken auf Eingangskanal B

© ifm electronic gmbh

5.2.6 Bausteine: Ausgangsfunktionen

Inhalt

CURRENT_CONTROL 181
OUTPUT 183
PWM1000 185
PWM1000_LOW 187
H_BRIDGE 189

15075
10462

Für dieses Gerät können Sie die Funktionsweise von einigen oder von allen Ausgängen einstellen. Hier zeigen wir Ihnen geeignete Bausteine dazu.

© ifm electronic gmbh

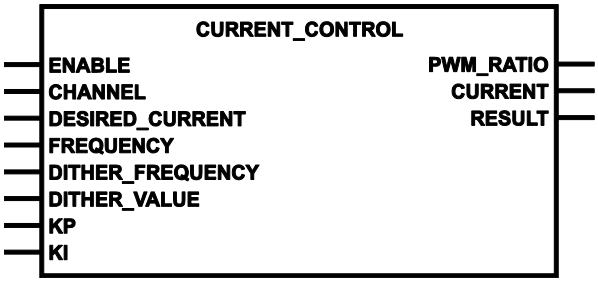
CURRENT_CONTROL

8082

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

8086

CURRENT_CONTROL arbeitet als Stromregler für die PWMi-Ausgänge.

Der Regler regelt in Abhängigkeit der Periodendauer des PWM-Signals. Die beiden Anstellparameter KI und KP repräsentieren den **I**ntegral- und den **P**roportional-Anteil des Reglers.

► Zur Ermittlung der besten Einstellung des Reglers bietet sich als Startwert an, KI=50 und KP=50 zu setzen. Je nach gewünschtem Reglerverhalten können die Werte schrittweise vergrößert (Regler wird härter / schneller) oder verkleinert (Regler wird schwächer / langsamer) werden.

> Bei Sollwert DESIRED_CURRENT=0 wird der Ausgang **sofort** auf 0 mA geschaltet, wobei **nicht** entsprechend der eingestellten Parameter auf 0 mA heruntergeregelt wird.

Der Regler verfügt über einen schnellen Ausgleichsmechanismus bei Spannungseinbrüchen der Versorgungsspannung. In Abhängigkeit der Größe des Spannungseinbruchs wird zusätzlich zum Regelverhalten des Reglers die Ratio des PWMs dementsprechend so vergrößert, dass der Regler so schnell wie möglich den Sollwert erreicht.

Je nach eingesetzter Steuerungs-Hardware ist ein unterschiedliches Teach-Verhalten zu beachten.

HINWEIS

► Bei der Definition des Parameters DITHER_VALUE darauf achten, dass das resultierende PWM-Ratio im Arbeitsbereich der Regelung zwischen 0...1000 ‰ bleibt:
- PWM-Ratio + DITHER_VALUE < 1000 ‰ und
- PWM-Ratio - DITHER_VALUE > 0 ‰.

Außerhalb dieses zulässigen Bereichs kann der im Parameter DESIRED_CURRENT angegebene Strom nicht erreicht werden.

> Bei PWM-Frequenzen unter 100 Hz und zusätzlichem Dither kann die Stromregelung die angegebene Genauigkeit (→ Datenblatt) nicht mehr erreichen.

© ifm electronic gmbh www.ifm.com

Parameter der Eingänge

17888

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > gestartete Prozesse laufen im Hintergrund weiter > FB-Ausgänge werden nicht aktualisiert
CHANNEL	BYTE	Nummer des stromgeregelten Ausgangskanals (0...7) 0...7 für die Ausgänge OUT0...OUT7
DESIRED_CURRENT	WORD	Stromsollwert des Ausgangs in [mA]
FREQUENCY	WORD	zulässige PWM-Frequenz am Ausgang in [Hz] zulässig = 20...250 = 0x0014...0x00FA
DITHER_FREQUENCY	WORD	Dither-Frequenz in [Hz] Wertebereich = 0...FREQUENCY / 2 FREQUENCY / DITHER_FREQUENCY muss geradzahlig sein! Alle anderen Werte erhöht der FB auf den nächst passenden Wert.
DITHER_VALUE	WORD	Spitze-Spitze-Wert des Dithers in [‰] zulässig = 0...1 000 = 0x0000...0x03E8
KP	BYTE	Proportional-Anteil des Ausgangsignals
KI	BYTE	Integral-Anteil des Ausgangsignals

Parameter der Ausgänge

8088

Parameter	Datentyp	Beschreibung
PWM_RATIO	WORD	Zu Kontrollzwecken: Anzeige PWM-Tastverhältnis 0...1000 ‰
CURRENT	WORD	nur für stromregelbare Ausgänge möglich: aktueller Ausgangs-Strom in [mA]
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
2	02	Funktionsbaustein ist aktiv (Aktion noch nicht beendet)
3	03	Funktionsbaustein ist aktiv – es liegen noch keine gültigen Werte vor
128	80	Unterspannung an VBBx
130	82	Kanaleinstellung ist ungültig
131	83	Wert für DESIRED_CURRENT ist ungültig
133	85	Wert für FREQUENCY ist ungültig
134	86	Dither-Einstellung ist ungültig

© ifm electronic gmbh www.ifm.com

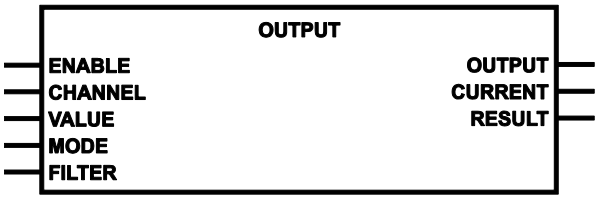
OUTPUT

8078

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

16288

OUTPUT weist einem Ausgangskanal eine Betriebsart zu (→ Datenblatt). Der FB ermöglicht die Zustandserfassung am gewählten Ausgangskanal.

Die Messung und der Ausgangswert resultieren aus der über MODE angegebenen Betriebsart:

- binärer Ausgang, plus-schaltend (BH) mit/ohne Diagnosefunktion
- binärer Ausgang, plus-schaltend (BH) mit Diagnosefunktion und Protection
- binärer Ausgang, minus-schaltend (BL) ohne Diagnosefunktion

Im laufenden Betrieb sollte die Betriebsart nicht geändert werden.

Parameter der Eingänge

16289

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CHANNEL	BYTE	Nummer des Ausgangskanals (0...7) 0...7 für die Ausgänge OUT0...OUT7
VALUE	BOOL	TRUE: Ausgang aktivieren FALSE: Ausgang deaktivieren
MODE	BYTE	Betriebsart des Bausteins: 0 = 0x00 = Aus 2 = 0x02 = Binärausgang, plus-schaltend 13 = 0x0D = Binärausgang, minus-schaltend 15 = 0x0F = Binärausgang, plus-schaltend mit Diagnose 16 = 0x10 = Binärausgang, plus-schaltend mit Diagnose und Protection
FILTER	BYTE	nur für Ausgänge mit Stromrücklesung: Filter für die Messung am Ausgang: zulässig = 0...8 empfohlen = 4 → Kapitel ***Software-Filter der Ausgänge konfigurieren*** (→ Seite 52) Für Ausgänge ohne Stromrücklesung: FILTER = 0 oder: Parameter FILTER nicht belegen!

Im laufenden Betrieb sollte die Betriebsart nicht geändert werden.

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

8081

Parameter	Datentyp	Beschreibung
OUTPUT	BOOL	TRUE: Ausgang ist aktiviert FALSE: Ausgang ist deaktiviert
CURRENT	WORD	nur für stromregelbare Ausgänge möglich: aktueller Ausgangs-Strom in [mA]
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
2	02	Funktionsbaustein ist aktiv (Aktion noch nicht beendet)
3	03	Funktionsbaustein ist aktiv – es liegen noch keine gültigen Werte vor
128	80	Unterspannung an VBBx
130	82	Kanaleinstellung ist ungültig
132	84	Moduseinstellung ist ungültig
136	88	Filtereinstellung ist ungültig
141	8D	ein Leiterbruch wurde erkannt (bei Binärausgang plus-schaltend (BH) mit Diagnose)
142	8E	ein Kurzschluss wurde erkannt (bei Binärausgang plus-schaltend (BH) mit Diagnose)
145	91	Strom am Ausgang ist zu hoch (bei Binärausgang plus-schaltend (BH) mit Diagnose und Protection)

© ifm electronic gmbh www.ifm.com

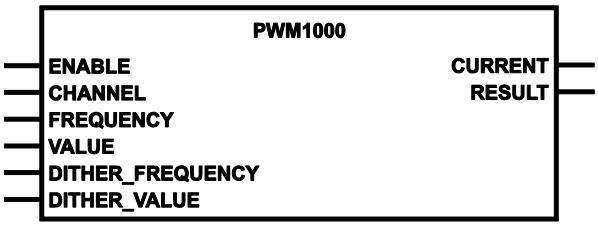
PWM1000

8060

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

8062

PWM1000 dient der Initialisierung und Parametrierung der PWM-Ausgänge.

Der FB ermöglicht eine einfache Anwendung der PWM-Funktion im Gerät. Für jeden Kanal kann jeweils eine eigene PWM-Frequenz und das Puls-Pause-Verhältnis eingestellt werden.

Die PWM-Frequenz FREQUENCY kann direkt in [Hz] und das Puls-Pause-Verhältnis VALUE in Schritten von 1 ‰ angegeben werden.

Parameter der Eingänge

8063

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > gestartete Prozesse laufen im Hintergrund weiter > FB-Ausgänge werden nicht aktualisiert
CHANNEL	BYTE	Nummer des PWM-Ausgangskanals (0...7) 0...7 für die Ausgänge OUT0...OUT7
FREQUENCY	WORD	PWM-Frequenz in [Hz] zulässig = 20...250 = 0x0014...0x00FA
VALUE	WORD	PWM-Wert (Puls-Periode-Verhältnis) in [‰] zulässig = 0...1 000 = 0x0000...0x03E8 Werte > 1 000 gelten als = 1 000
DITHER_FREQUENCY	WORD	Dither-Frequenz in [Hz] Wertebereich = 0...FREQUENCY / 2 FREQUENCY / DITHER_FREQUENCY muss geradzahlig sein! Alle anderen Werte erhöht der FB auf den nächst passenden Wert.
DITHER_VALUE	WORD	Spitze-Spitze-Wert des Dithers in [‰] zulässig = 0...1 000 = 0x0000...0x03E8

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

8523

Parameter	Datentyp	Beschreibung
CURRENT	WORD	nur für stromregelbare Ausgänge möglich: aktueller Ausgangs-Strom in [mA]
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
2	02	Funktionsbaustein ist aktiv (Aktion noch nicht beendet)
3	03	Funktionsbaustein ist aktiv – es liegen noch keine gültigen Werte vor
128	80	Unterspannung an VBBx
130	82	Kanaleinstellung ist ungültig
131	83	Wert für VALUE ist ungültig
132	84	Moduseinstellung ist ungültig
133	85	Wert für FREQUENCY ist ungültig
134	86	Dither-Einstellung ist ungültig

© ifm electronic gmbh www.ifm.com

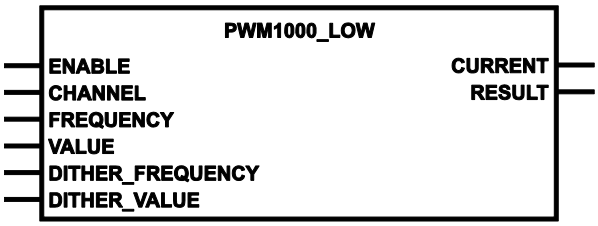
PWM1000_LOW

16399

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

16408

PWM1000_LOW dient der Initialisierung und Parametrierung der minus-schaltenden PWM-Ausgänge.

Der FB ermöglicht eine einfache Anwendung der PWM-Funktion im Gerät. Für jeden Kanal kann jeweils eine eigene PWM-Frequenz und das Puls-Pause-Verhältnis eingestellt werden.

Die PWM-Frequenz FREQUENCY kann direkt in [Hz] und das Puls-Pause-Verhältnis VALUE in Schritten von 1 ‰ angegeben werden.

Parameter der Eingänge

17886

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > gestartete Prozesse laufen im Hintergrund weiter > FB-Ausgänge werden nicht aktualisiert
CHANNEL	BYTE	Nummer des PWM-Ausgangskanals (4...7) 4...7 für die Ausgänge OUT4...OUT7
FREQUENCY	WORD	PWM-Frequenz in [Hz] zulässig = 20...250 = 0x0014...0x00FA
VALUE	WORD	PWM-Wert (Puls-Periode-Verhältnis) in [‰] zulässig = 0...1 000 = 0x0000...0x03E8 Werte > 1 000 gelten als = 1 000
DITHER_FREQUENCY	WORD	Dither-Frequenz in [Hz] Wertebereich = 0...FREQUENCY / 2 FREQUENCY / DITHER_FREQUENCY muss geradzahlig sein! Alle anderen Werte erhöht der FB auf den nächst passenden Wert.
DITHER_VALUE	WORD	Spitze-Spitze-Wert des Dithers in [‰] zulässig = 0...1 000 = 0x0000...0x03E8

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

16273

Parameter	Datentyp	Beschreibung
CURRENT	WORD	immer = 0 im minus-schaltenden PWM-Pfad ist keine Strommessung verfügbar
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
2	02	Funktionsbaustein ist aktiv (Aktion noch nicht beendet)
3	03	Funktionsbaustein ist aktiv – es liegen noch keine gültigen Werte vor
128	80	Unterspannung an VBBx
130	82	Kanaleinstellung ist ungültig
131	83	Wert für VALUE ist ungültig
132	84	Kanaleinstellung ist ungültig gewählter minus-schaltender Kanal erfüllt Anforderung nicht
133	85	Wert für FREQUENCY ist ungültig
134	86	Dither-Einstellung ist ungültig

© ifm electronic gmbh www.ifm.com

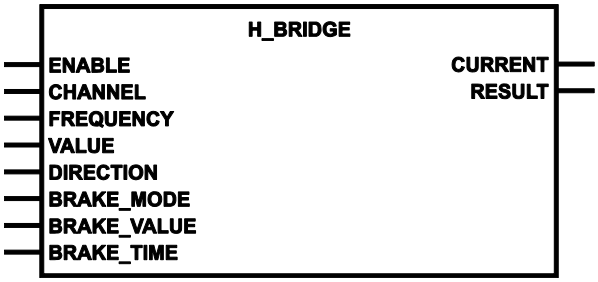
H_BRIDGE

16233

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_util_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

16412

H_BRIDGE organisiert das Ansteuern der H-Brücken an den PWM-Kanälen.

Der FB dient zur einfachen Verwendung der Ausgänge als H-Brücke. Dazu werden jeweils zwei aufeinander folgende Ausgangskanäle mit minus-schaltendem Treiber zu einer Brücke zusammengefasst.

Drehrichtung festlegen: Ist DIRECTION = FALSE, wird beim ersten Ausgang der plus-schaltende Treiber über ein PWM-Signal angesteuert und der minus-schaltende Treiber des zweiten Ausgangs ist durchgeschaltet.

Bremsmethode festlegen:

BRAKE_MODE	Bremsverhalten	Wende-Verhalten
0	Kein Bremsen (entspricht VALUE = 0) Der Antrieb "trudelt" aus	Kein Bremsen Die Drehrichtung wechselt sofort
1	Maximale Bremswirkung (Notbremse) Falls VALUE = 0: der Antrieb bremst während und nach Ablauf von BRAKE_TIME	Maximale Bremswirkung (Notbremse) während BRAKE_TIME Nach Ablauf von BRAKE_TIME wechselt die Drehrichtung
2	Maximale Bremswirkung (Notbremse) während BRAKE_TIME Nach Ablauf von BRAKE_TIME "trudelt" der Antrieb aus	Maximale Bremswirkung (Notbremse) während BRAKE_TIME Nach Ablauf von BRAKE_TIME wechselt die Drehrichtung
3	Antrieb bremst mit BRAKE_VALUE Falls VALUE = 0: der Antrieb bremst während und nach Ablauf von BRAKE_TIME	Antrieb bremst mit BRAKE_VALUE während BRAKE_TIME Nach Ablauf von BRAKE_TIME wechselt die Drehrichtung
4	Antrieb bremst mit BRAKE_VALUE während BRAKE_TIME Nach Ablauf von BRAKE_TIME "trudelt" der Antrieb aus	Antrieb bremst mit BRAKE_VALUE während BRAKE_TIME Nach Ablauf von BRAKE_TIME wechselt die Drehrichtung

© ifm electronic gmbh www.ifm.com

18938

ACHTUNG

Gefahr von thermischer Überlastung!

Bei Nutzung der Notbremse (BRAKE_MODE = 1...2) können der Antrieb und / oder die Ausgangstreiber der Steuerung wegen thermischer Überlastung beschädigt werden!
Grenzwert für die maximal rückspeisbare Energie → Datenblatt "Max. Energierückspeisung"

► Den Antrieb mit BRAKE_MODE = 3...4 sanft abbremsen.

HINWEIS

Bei Einsatz der H-Brücke wird die Stromregelung nicht unterstützt.

Ausgänge, die im PWM-Modus betrieben werden, unterstützen keine Diagnosefunktionen.

► FB in jedem SPS-Zyklus aufrufen!

Lage der als H-Brücke verwendbaren Ausgangskanäle → Datenblatt.

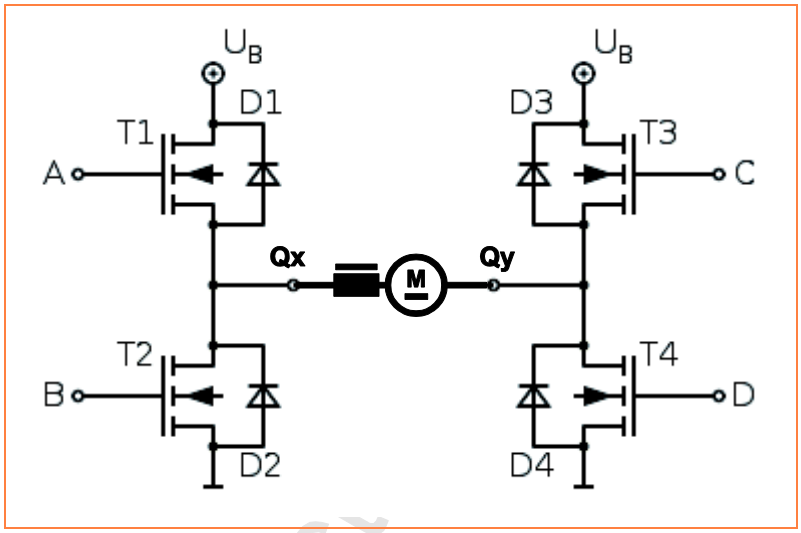
Der FB benötigt die in der gleichen Bibliothek vorhandenen Funktionen.
Diese Funktionen nicht separat einsetzen!

Prinzip der H-Brücke

16279
16411

Hier sehen Sie, wie eine H-Brücke am ifm-Controller via PWM-Ausgängen betrieben werden kann.

Prinzipschaltung einer H-Brücke mit PWM-Ansteuerung:

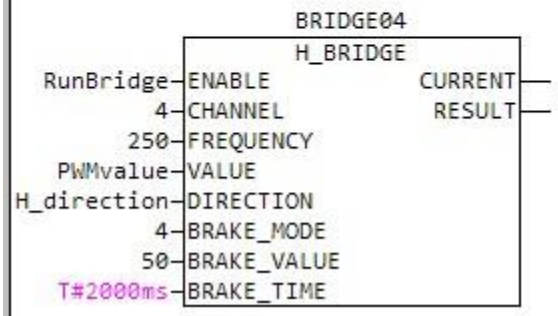


T1 und T2 bilden zusammen z.B. den Ausgang Qx.
Genauso bilden T3 und T4 z.B. den Ausgang Qy.
Dadurch werden nur zwei Anschlüsse für den DC-Motor benötigt.

© ifm electronic

Programm-Beispiel:

```
VAR
    BRIDGE04: H_BRIDGE;         (* H-bridge for output channels 04/05          *)
    RunBridge: BOOL := FALSE;   (* TRUE = run the H-bridge                     *)
    PWMvalue: WORD := 100;      (* current PWM value (0...1000)                *)
    H_direction: BOOL := TRUE; (* TRUE = counter clockwise; FALSE = clockwise *)
END_VAR
```



© ifm electronic gmbh

Parameter der Eingänge

6885

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > gestartete Prozesse laufen im Hintergrund weiter > FB-Ausgänge werden nicht aktualisiert
CHANNEL	BYTE	Erster Kanal des Ausgangspaares: CHANNEL = 4 ⇨ OUT04+OUT05 CHANNEL = 6 ⇨ OUT06+OUT07
FREQUENCY	WORD	PWM-Frequenz in [Hz] zulässig = 20...250 = 0x0014...0x00FA
VALUE	WORD	PWM-Wert (Puls-Periode-Verhältnis) in [‰] zulässig = 0...1 000 = 0x0000...0x03E8 Werte > 1 000 gelten als = 1 000
DIRECTION	BOOL	Drehrichtung des Motors: TRUE: PWM plus-schaltend (BH) ist auf Kanal B FALSE: PWM plus-schaltend (BH) ist auf Kanal A
BRAKE_MODE	BYTE	Bremsmethode für den Antrieb an der H-Brücke:
		0: Kein Bremsen (entspricht VALUE = 0) Der Antrieb "trudelt" aus
		1: Maximale Bremswirkung (Notbremse) Falls VALUE = 0: der Antrieb bremst während und nach Ablauf von BRAKE_TIME
		2: Maximale Bremswirkung (Notbremse) während BRAKE_TIME Nach Ablauf von BRAKE_TIME "trudelt" der Antrieb aus
		3: Antrieb bremst mit BRAKE_VALUE Falls VALUE = 0: der Antrieb bremst während und nach Ablauf von BRAKE_TIME
		4: Antrieb bremst mit BRAKE_VALUE während BRAKE_TIME Nach Ablauf von BRAKE_TIME "trudelt" der Antrieb aus
		1…4: Wenn Drehrichtungswechsel angefordert: Nach Ablauf von BRAKE_TIME wechselt die Drehrichtung
BRAKE_VALUE	WORD	PWM-Wert in [‰] für den minus-schaltenden Ausgang beim Bremsen
BRAKE_TIME	TIME	Bremsdauer des Antriebs an der H-Brücke zulässig: 0...3 600 s

© ifm electronic gmbh

Parameter der Ausgänge

6905

Parameter	Datentyp	Beschreibung
CURRENT	WORD	nur für Ausgänge mit Strommessung möglich: aktueller Ausgangs-Strom in [mA] Bei Bremsbetrieb ist keine Strommessung verfügbar!
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
2	02	Funktionsbaustein ist aktiv (Aktion noch nicht beendet)
3	03	Funktionsbaustein ist aktiv – es liegen noch keine gültigen Werte vor
128	80	Unterspannung an VBBx
130	82	Kanaleinstellung ist ungültig Kanal existiert nicht oder hat ungerade Kanalnummer
131	83	Wert für VALUE ist ungültig
132	84	Kanaleinstellung ist ungültig gewählter minus-schaltender Kanal erfüllt Anforderung nicht
133	85	Wert für FREQUENCY ist ungültig
140	8C	Wert für BRAKE_TIME nicht im zulässigen Bereich

© ifm electronic gmbh www.ifm.com

5.2.7 Bausteine: System

Inhalt

FLASH_INFO 195
FLASH_READ 196
GET_APP_INFO 197
GET_HW_INFO 198
GET_IDENTITY 199
GET_SW_INFO 200
GET_SW_VERSION 201
MEM_ERROR 202
MEMCPY 203
OHC 205
SET_IDENTITY 207
SET_LED 208
SET_PASSWORD 210
TIMER_READ_US 211

15067

Hier zeigen wir Ihnen ifm-Funktionen, mit denen Sie Folgendes erreichen:
- Speicherinhalte verwalten
- Informationen von Software und Hardware lesen
- diverse Daten und Parameter setzen oder lesen

© ifm electronic gmbh

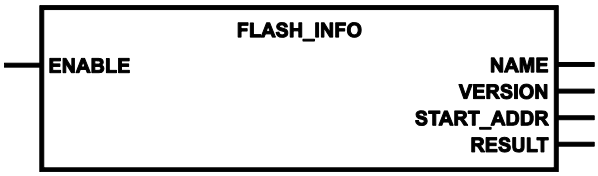
FLASH_INFO

11580

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

11588

FLASH_INFO liest die Informationen aus dem User-Flash-Speicher:
- Name des Speicherbereichs (vom User vorgegeben),
- Software-Version,
- Startadresse (für einfaches Lesen mit IEC-Struktur).

Parameter der Eingänge

11589

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > gestartete Prozesse laufen im Hintergrund weiter > FB-Ausgänge werden nicht aktualisiert

Parameter der Ausgänge

11590

Parameter	Datentyp	Beschreibung
NAME	STRING(24)	Name des Speicherbereichs (vom User vorgegeben)
VERSION	STRING(24)	Software-Version
START_ADDR	DWORD	Startadresse der Daten
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
157	9D	Software-Header ist ungültig (CRC-Fehler)

© ifm electronic gmbh www.ifm.com

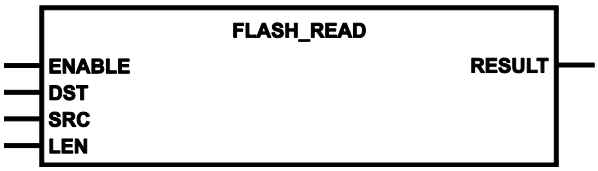
FLASH_READ

8147

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

11579

FLASH_READ ermöglicht das Lesen unterschiedlicher Datentypen direkt aus dem Flash-Speicher.

Der FB liest den Inhalt ab der Adresse von SRC aus dem Flash-Speicher. Dabei werden genau so viele Bytes übertragen, wie diese unter LEN angegeben sind.

- ► Die aus SRC + LEN resultierende Adresse muss ≤ 65 408 sein.
- ► Für die Zieladresse DST gilt:
  Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

Parameter der Eingänge

8148

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
DST	DWORD	Startadresse im Zielspeicher Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!
SRC	DWORD	relative Anfangsadresse im Speicher zulässig = 0...65 407 = 0x0000 0000...0x0000 FF7F
LEN	WORD	Anzahl (≥ 1) der zu übertragenden Daten-Bytes

Parameter der Ausgänge

8152

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
152	98	unzulässiger Speicherbereich: • ungültige Quell-Adresse • ungültige Ziel-Adresse • ungültige Anzahl Bytes

© ifm electronic gmbh www.ifm.com

GET_APP_INFO

11581

= Get Application Information

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:

Beschreibung

11593

GET_APP_INFO liefert Informationen über das im Gerät gespeicherte Anwendungsprogramm:
- Name (= Dateiname des CODESYS-Projekts),
- Version (= aus CODESYS-Menü [Projekt] > [Projektinformation] > [Version]),
- eindeutige CODESYS-Build-Nummer,
- CODESYS-Build-Datum.

Parameter der Eingänge

11594

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert

Parameter der Ausgänge

11595

Parameter	Datentyp	Beschreibung
NAME	STRING(24)	Name der Anwendung
VERSION	STRING(24)	Version des Anwendungsprogramms
BUILD_NUM	STRING(24)	eindeutige CODESYS-Build-Nummer (z.B.: "45")
BUILD_DATE	STRING(24)	CODESYS-Build-Datum (z.B.: "20111006123800")
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig

© ifm electronic gmbh www.ifm.com

GET_HW_INFO

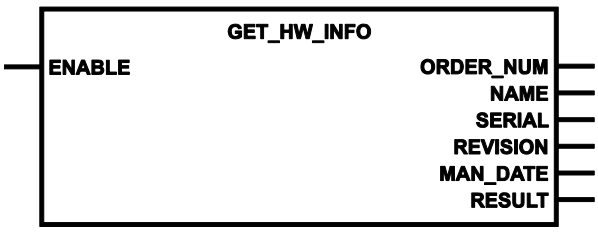
11582

= Get Hardware Information

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

1599

GET_HW_INFO liefert Informationen über die Hardware des Geräts:
- ifm-Artikelnummer (z.B. CR0403),
- Artikelbezeichnung,
- eindeutige Seriennummer,
- Hardware-Revision,
- Produktionsdatum.

Parameter der Eingänge

11600

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert

Parameter der Ausgänge

11601

Parameter	Datentyp	Beschreibung
ORDER_NUM	STRING(24)	ifm-Artikelnummer (z.B.: CR0403)
NAME	STRING(24)	Artikelbezeichnung (z.B.: "BasicController 12/12")
SERIAL	STRING(24)	Seriennummer des Geräts (z.B.: "000045784")
REVISION	STRING(24)	Hardware-Revisionsstand des Geräts (z.B.: "V01.00.01")
MAN_DATE	STRING(24)	Herstellungsdatum des Geräts (z.B.: "20111007123800")
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig

© ifm electronic gmbh www.ifm.com

GET_IDENTITY

8166

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:

Beschreibung

15411

GET_IDENTITY liest die im Gerät gespeicherte Kennung (wurde zuvor mit ***SET_IDENTITY*** (→ Seite 207) gespeichert).

Parameter der Eingänge

8167

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert

Parameter der Ausgänge

8168

Parameter	Datentyp	Beschreibung
APP_IDENT	STRING(80)	Kennung der Anwendung als Zeichenkette von max. 80 Zeichen, z.B.: "Crane1704"
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
155	9B	Wert konnte nicht gelesen werden

© ifm electronic gmbh www.ifm.com

GET_SW_INFO

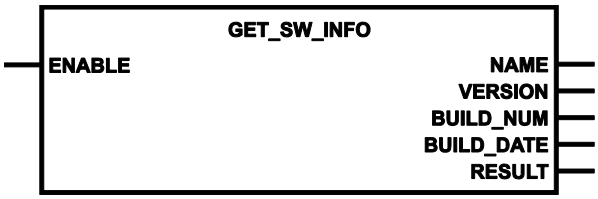
11583

= Get Software Information

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

11596

GET_SW_INFO liefert Informationen über die System-Software des Geräts:
- Software-Name,
- Software-Version,
- Build-Nummer,
- Build-Datum.

Parameter der Eingänge

11597

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert

Parameter der Ausgänge

11598

Parameter	Datentyp	Beschreibung
NAME	STRING(24)	Name der System-Software (z.B.: "BasicSystem")
VERSION	STRING(24)	Version der System-Software (z.B.: "V02.00.03")
BUILD_NUM	STRING(24)	Build-Nummer der System-Software (z.B.: "45")
BUILD_DATE	STRING(24)	Build-Datum der System-Software (z.B.: "20111006123800")
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig

© ifm electronic gmbh www.ifm.com

GET_SW_VERSION

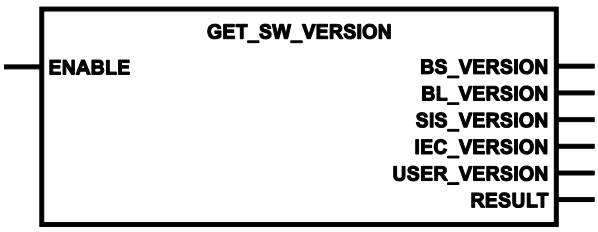
14763

= Get Software-Version

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

14765

GET_SW_VERSION liefert die Informationen zur Software im Gerät:
- BasicSystem-Version
- Bootloader-Version
- SIS-Version
- IEC-Anwendungsprogramm-Version
- IEC-User-Flash-Version

Parameter der Eingänge

14766

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert

Parameter der Ausgänge

14767

Parameter	Datentyp	Beschreibung
BS_VERSION	STRING(24)	Basic-System-Version
BL_VERSION	STRING(24)	Bootloader-Version
SIS_VERSION	STRING(24)	SIS-Version (SIS = **S**ystem **I**nformation **S**ervice)
IEC_VERSION	STRING(24)	IEC-Anwendungsprogramm-Version
USER_VERSION	STRING(24)	IEC-User-Flash-Version
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig

© ifm electronic gmbh www.ifm.com

MEM_ERROR

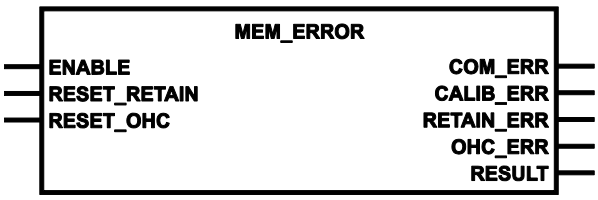
14770

= Memory Error

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

14772

MEM_ERROR meldet Fehler in einigen Parametern oder im Speicher.
Über entsprechende FB-Eingänge lassen sich die Speicherbereiche löschen.

Parameter der Eingänge

14773

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
RESET_RETAIN	BOOL	TRUE: Remanenten Retain-Speicher löschen FALSE: keine Änderung am Speicherinhalt
RESET_OHC	BOOL	TRUE: Remanenten OHC-Speicher löschen FALSE: keine Änderung am Speicherinhalt

Parameter der Ausgänge

14774

Parameter	Datentyp	Beschreibung
COM_ERR	BOOL	Download ID und Baudrate sind auf Voreinstellwerte gesetzt (Download-Parameter gingen verloren)
CALIB_ERR	BOOL	Kalibrierungswerte sind ungültig (analoge Eingänge, PWM-Ausgänge, Systemspannungen)
RETAIN_ERR	BOOL	Retain-Speicher ist ungültig (z.B. wegen zu starken Magnetfelds teilweise gelöscht)
OHC_ERR	BOOL	Betriebsstundenzähler OHC Werte sind ungültig (z.B. wegen zu starken Magnetfelds teilweise gelöscht)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig

© ifm electronic gmbh www.ifm.com

MEMCPY

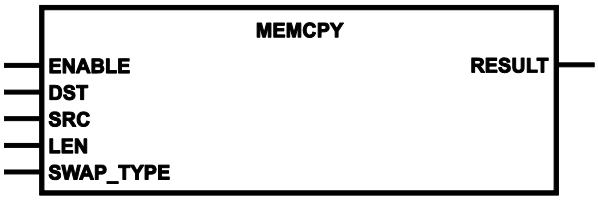
8160

= Memory Copy

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

412

MEMCPY ermöglicht das Schreiben und Lesen unterschiedlicher Datentypen direkt in den Speicher.

Der FB schreibt den Inhalt ab der Adresse von SRC an die Adresse DST.

► Für die Adressen SRC und DST gilt:
Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

> Dabei werden genau so viele Bytes übertragen, wie diese unter LEN angegeben wurden. Dadurch ist es auch möglich, genau ein Byte einer Word-Variablen zu übertragen.

Parameter der Eingänge

8162

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
DST	DWORD	Startadresse im Zielspeicher Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!
SRC	DWORD	Quell-Adresse
LEN	WORD	Anzahl der zu kopierenden Bytes SWAP_TYPE=1: LEN muss Mehrfaches von 2 sein SWAP_TYPE=2: LEN muss Mehrfaches von 4 sein
SWAP_TYPE	BYTE	Byte-Reihenfolge tauschen: 0 = kein Tausch z.B.: 1A 2B 3C 4D ⇨ 1A 2B 3C 4D 1 = 2 Bytes tauschen (WORD, INT, ...) z.B.: 1A 2B 3C 4D ⇨ 2B 1A 4D 3C LEN muss ein Mehrfaches von 2 sein! 2 = 4 Bytes tauschen (DWORD, DINT, REAL, TIME, ...) z.B.: 1A 2B 3C 4D ⇨ 4D 3C 2B 1A LEN muss ein Mehrfaches von 4 sein!

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

8163

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
152	98	unzulässiger Speicherbereich: • ungültige Quell-Adresse • ungültige Ziel-Adresse • ungültige Anzahl Bytes
156	9C	unzulässige Werte: • ungültiger Wert für SWAP_TYPE • LEN passt nicht zu SWAP_TYPE

© ifm electronic gmbh www.ifm.com

OHC

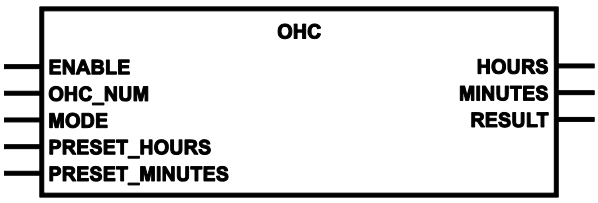
14777

= Operating Hours Counter

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

14778

OHC bietet 4 universell verwendbare Betriebsstundenzähler.
Aber bei Hardware-Stand < AD: nur 2 Betriebsstundenzähler möglich.

Gültiger Zählbereich: 0:00...4 294 967 295:59 Stunden (= 490 293 Jahre, 25 Tage, 15 Stunden)

Falls Hardware-Stand des Geräts < AD:
Den Speicherbereich für OHC einmalig zurücksetzen:
▶ Im FB ***MEM_ERROR*** (→ Seite 202) den Eingang RESET_OHC = TRUE setzen!
> Erst jetzt sind die Betriebsstundenzähler verwendbar.

Parameter der Eingänge

14779

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > gestartete Prozesse laufen im Hintergrund weiter > FB-Ausgänge werden nicht aktualisiert
OHC	BYTE	Operating Hours Counter = Betriebsstundenzähler Nummer des Zählers (0...3)
MODE	BYTE	Betriebsmodus des Zählers Zulässige Werte = 0 = Zähler anhalten 1 = Zählen fortsetzen beim zuletzt gespeicherten Wert 2 = Zähler zurücksetzen 3 = Zähler voreinstellen mit den folgenden Werten
PRESET_HOURS	DWORD	Voreinstellwert Stunden (0...4 294 967 295 = 0x0000 0000...0xFFFF FFFF)
PRESET_MINUTES	BYTE	Voreinstellwert Minuten (0...59 = 0x00...0x3B)

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

14780

Parameter	Datentyp	Beschreibung
HOURS	DWORD	Zählerstand Stunden (0...4 294 967 295 = 0x0000 0000...0xFFFF FFFF)
MINUTES	BYTE	Zählerstand Minuten (0...59 = 0x00...0x3B)
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
130	82	Zählernummer in OHC_NUM ist ungültig
131	83	Voreinstellwert ist ungültig
132	84	Moduseinstellung ist ungültig
158	9E	Remanent-Speicher ist ungültig (CRC-Fehler)

© ifm electronic gmbh www.ifm.com

SET_IDENTITY

8174

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:

Beschreibung

8535

SET_IDENTITY setzt eine anwendungsspezifische Programmkennung.

Mit dem FB kann durch das Anwendungsprogramm eine Programmkennung erzeugt werden.

► Diese Kennung kann zur Identifizierung des geladenen Programms ausgelesen werden:
- über das Maintenance-Tool
- im Anwendungsprogramm über den FB ***GET_IDENTITY*** (→ Seite 199)

Parameter der Eingänge

8175

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
APP_IDENT	STRING(80)	Kennung der Anwendung als Zeichenkette von max. 80 Zeichen, z.B.: "Crane1704" Rücksetzen mit APP_IDENT = ""

Parameter der Ausgänge

8176

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet

© ifm electronic gmbh www.ifm.com

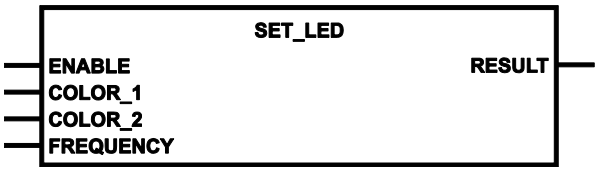
SET_LED

8052

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

8054

Mit SET_LED können im Anwendungsprogramm Frequenz und Farbe der Status-LED geändert werden.

Wird der Blinkmodus im Anwendungsprogramm geändert, gilt die Voreinstellung-Tabelle nicht mehr (→ Kapitel ***Status-LED*** (→ Seite 25)).

Parameter der Eingänge

8223

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
COLOR_1	BYTE	LED-Farbe für "eingeschaltet" Farbkonstante aus der Datenstruktur "System LED Color" zulässige Werte: 00 = LED_BLACK (= LED aus) 01 = LED_RED 02 = LED_GREEN 03 = LED_YELLOW
COLOR_2	BYTE	LED-Farbe für "ausgeschaltet" Farbkonstante aus der Datenstruktur "System LED Color" zulässige Werte: 00 = LED_BLACK (= LED aus) 01 = LED_RED 02 = LED_GREEN 03 = LED_YELLOW
FREQUENCY	BYTE	LED-Blinkfrequenz Frequenzkonstante aus der Datenstruktur "System LED Frequency"; zulässige Werte: 00 = LED_0HZ = dauernd EIN 01 = LED_05HZ = blinkt mit 0,5 Hz 02 = LED_1Hz = blinkt mit 1 Hz 04 = LED_2HZ = blinkt mit 2 Hz 10 = LED_5HZ = blinkt mit 5 Hz

© ifm electronic gmbh www.ifm.com

Parameter der Ausgänge

8227

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet
2	02	Funktionsbaustein ist aktiv (Aktion noch nicht beendet)
133	85	Wert für FREQUENCY ist ungültig
151	97	Wert für Farbe ist ungültig

© ifm electronic gmbh www.ifm.com

SET_PASSWORD

8178

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:

Beschreibung

8179

SET_PASSWORD setzt Benutzerkennung für Programm- und Speicher-Upload mit dem Maintenance-Tool.

Ist die Benutzerkennung aktiv, kann durch das Maintenance-Tool das Anwendungsprogramm oder der Datenspeicher nur ausgelesen werden, wenn das richtige Passwort eingegeben wurde.

Wird an den Eingang PASSWORD ein Leer-String (Default-Zustand) übergeben, so wird ein zuvor gesetztes Passwort zurückgesetzt. Ein Upload des Anwendungsprogramms oder des Datenspeichers ist dann jederzeit möglich.

> Beim Laden eines neuen Anwendungsprogramms wird das Passwort wieder zurückgesetzt.

Parameter der Eingänge

8180

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
PASSWORD	STRING(16)	Benutzerkennung Wenn PASSWORD = "", dann ist Zugriff ohne Passworteingabe möglich.

Parameter der Ausgänge

8181

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet

© ifm electronic gmbh www.ifm.com

TIMER_READ_US

8219

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0411_Vxxyyzz.LIB`

Symbol in CODESYS:

Beschreibung

660

TIMER_READ_US liest die aktuelle Systemzeit in [µs] aus.

Mit Anlegen der Versorgungsspannung bildet das Gerät einen Zeittakt, der in einem Register aufwärts gezählt wird. Dieses Register kann mittels des FB-Aufrufes ausgelesen werden und z.B. zur Zeitmessung genutzt werden.

> **Info**
>
> Der System-Timer läuft maximal bis zum Zählerwert 1h 11min 34s 967ms 295µs und startet anschließend wieder mit 0.

Parameter der Ausgänge

8220

Parameter	Datentyp	Beschreibung
TIME_US	DWORD	Aktuelle Systemzeit [µs]
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Funktionsbaustein-Ausführung ohne Fehler beendet

© ifm electronic gmbh www.ifm.com

6 Diagnose und Fehlerbehandlung

Inhalt

Diagnose 212
Fehler 212
Reaktion auf Fehlermeldungen 213
CAN / CANopen: Fehler und Fehlerbehandlung 213

19598

Das Laufzeitsystem (LZS) überprüft das Gerät durch interne Fehler-Checks:
• in der Startphase (Reset-Phase)
• während der Ausführung des Anwenderprogramms

So wird eine möglichst hohe Betriebssicherheit gewährleistet.

6.1 Diagnose

19601

Bei der Diagnose wird der "Gesundheitszustand" des Gerätes geprüft. Es soll festgestellt werden, ob und gegebenenfalls welche →Fehler im Gerät vorhanden sind.

Je nach Gerät können auch die Ein- und Ausgänge auf einwandfreie Funktion überwacht werden:
- Drahtbruch,
- Kurzschluss,
- Wert außerhalb des Sollbereichs.

Zur Diagnose können Konfigurations-Dateien herangezogen werden, die während des "normalen" Betriebs des Gerätes erzeugt wurden.
Der korrekte Start der Systemkomponenten wird während der Initialisierungs- und Startphase überwacht.
Zur weiteren Diagnose können auch Selbsttests durchgeführt werden.

6.2 Fehler

19602

Ein Fehler ist die Unfähigkeit einer Einheit, eine geforderte Funktion auszuführen.
Kein Fehler ist diese Unfähigkeit während vorbeugender Wartung oder anderer geplanter Handlungen oder aufgrund des Fehlers externer Mittel.
Ein Fehler ist oft das Resultat eines Ausfalls der Einheit selbst, kann aber ohne vorherigen Ausfall bestehen.
In der ISO 13849-1 ist mit "Fehler" der "zufällige Fehler" gemeint.

© ifm electronic gmbh

6.3 Reaktion auf Fehlermeldungen

8504

Es liegt grundsätzlich in der Verantwortung des Programmierers, auf die Fehlermeldungen im Anwendungsprogramm zu reagieren. Über die Fehlermeldung erhält man eine Fehlerbeschreibung.

> Das System setzt die Fehlermeldung zurück, sobald der fehlerauslösende Zustand nicht mehr vorliegt.

6.3.1 Beispielablauf für Reaktion auf Fehlermeldungen

8505

Das Laufzeitsystem schreibt zyklisch den Systemmerker TEMPERATURE.

Das Anwendungsprogramm erkennt die Gerätetemperatur durch Abfrage der INT-Variable.
Falls für die Anwendung zulässige Werte über- oder unterschritten werden:

> Das Anwendungsprogramm schaltet die Ausgänge ab.

► Die Ursache des Fehlers beheben.

> Anwendungsprogramm erkennt den wieder normalen Temperaturwert:
Die Maschine / Anlage darf neu gestartet oder weiter betrieben werden.

6.4 CAN / CANopen: Fehler und Fehlerbehandlung

19604

→ Systemhandbuch "Hintergrundwissen ecomat*mobile*"
→ Kapitel ***CAN / CANopen: Fehler und Fehlerbehandlung***

© ifm electronic gmbh www.ifm.com

7 Anhang

Inhalt

Systemmerker 214
Adressbelegung und E/A-Betriebsarten 214
Fehler-Tabellen 218

1664

Hier stellen wir Ihnen – ergänzend zu den Angaben in den Datenblättern – zusammenfassende Tabellen zur Verfügung.

7.1 Systemmerker

8374

8440

Die zu den Systemmerkern gehörenden Merkeradressen können sich bei einer Erweiterung der Steuerungskonfiguration ändern.

► Für die Programmierung nur die Symbolnamen der Systemmerker nutzen!

(→ Kapitel *Fehler-Codes und Diagnoseinformationen*)

Systemmerker (Symbolname)	Typ	Beschreibung
TEMPERATURE	INT	Temperatur im Gerät [°C]
SUPPLY_VOLTAGE_VBBS	WORD	Versorgungsspannung an VBBS in [mV]
SUPPLY_VOLTAGE_VBBx	WORD	Versorgungsspannung an VBBx in [mV] CR040n: x = 1, 2 CR041n: x = 1, 2 CR253n: x = 1, 2
SUPPLY_VOLTAGE_VU	WORD	interne Versorgungsspannung in [mV]

7.2 Adressbelegung und E/A-Betriebsarten

Inhalt

Adressbelegung Ein-/Ausgänge 215
Mögliche Betriebsarten Ein-/Ausgänge 216

1656

→ auch Datenblatt

© ifm electronic gmbh

7.2.1 Adressbelegung Ein-/Ausgänge

Inhalt

Eingänge: Adressbelegung 215
Ausgänge: Adressbelegung 215

2371

Eingänge: Adressbelegung

16654

Abkürzungen → Kapitel ***Hinweise zur Anschlussbelegung*** (→ Seite 24)
Betriebsarten der Ein- und Ausgänge → Kapitel ***Mögliche Betriebsarten Ein-/Ausgänge*** (→ Seite 216)

IEC-Adresse	Symbolische Adresse
%IB00	IN00
%IB01	IN01
%IB02	IN02
%IB03	IN03
%IB04	IN04
%IB05	IN05
%IB06	IN06
%IB07	IN07

Ausgänge: Adressbelegung

16468

Abkürzungen → Kapitel ***Hinweise zur Anschlussbelegung*** (→ Seite 24)
Betriebsarten der Ein- und Ausgänge → Kapitel ***Mögliche Betriebsarten Ein-/Ausgänge*** (→ Seite 216)

IEC-Adresse	Symbolische Adresse
%QB0	OUT00
%QB1	OUT01
%QB2	OUT02
%QB3	OUT03
%QB4	OUT04
%QB5	OUT05
%QB6	OUT06
%QB7	OUT07

© ifm electronic gmbh www.ifm.com

7.2.2 Mögliche Betriebsarten Ein-/Ausgänge

Inhalt

Eingänge: Betriebsarten 216
Ausgänge: Betriebsarten 217

2386

Eingänge: Betriebsarten

16655

= diese Konfiguration ist voreingestellt

Eingänge	mögliche Betriebsart		einstellen mit FB	FB-Eingang	Wert	
					dez	hex
IN00…IN03	Aus		INPUT	MODE	0	00
	Spannungseingang	0…10 000 mV	INPUT	MODE	3	03
	Spannungseingang ratiometrisch	0…1 000 ‰	INPUT	MODE	6	06
	Stromeingang	0…20 000 µA	INPUT	MODE	7	07
	Spannungseingang	0…32 000 mV	INPUT	MODE	9	09
	Binärer Eingang	plus-schaltend	INPUT	MODE	10	0A
	Binärer Eingang mit Diagnose (Namur)	plus-schaltend	INPUT	MODE	11	0B
	Binärer Eingang	minus-schaltend	INPUT	MODE	12	0C
	Widerstandseingang	16…30 000 Ohm	INPUT	MODE	18	12
	Frequenzmessung	0…30 000 Hz	PERIOD	MODE	14	0E
	Periodendauermessung	0,1...3 000 Hz	PERIOD	MODE	19	13
	Periodendauer- und Ratiomessung	0,1...3 000 Hz	PERIOD	MODE	20	14
	Phasenverschiebung	0...359°	PERIOD	MODE	25	19
	Aufwärtszähler Abwärtszähler	0…30 000 Hz	FASTCOUNT	MODE	21 22	15 16
	Drehgeber erfassen	0…1 000 Hz	INC_ENCODER			
IN04…IN07	Aus		INPUT	MODE	0	00
	Binärer Eingang	plus-schaltend	INPUT	MODE	10	0A
	Binärer Eingang mit Diagnose (Namur)	plus-schaltend	INPUT	MODE	11	0B
	Widerstandseingang	16…30 000 Ohm	INPUT	MODE	18	12

Betriebsarten mit folgendem Funktionsbaustein einstellen:

INPUT (→ Seite 176)	weist einem Eingangskanal eine Betriebsart zu liefert den aktuellen Zustand am gewählten Kanal
FASTCOUNT (→ Seite 171)	Zählerbaustein für schnelle Eingangsimpulse
INC_ENCODER (→ Seite 173)	Vorwärts-/Rückwärts-Zählerfunktion zur Auswertung von Drehgebern
PERIOD (→ Seite 178)	• misst am angegebenen Kanal: die Frequenz und die Periodendauer (Zykluszeit) in [µs], • misst am angegebenen Kanalpaar: die Phasenverschiebung in [°] zwischen Kanal A und Kanal B

© ifm electronic gmbh www.ifm.com

Ausgänge: Betriebsarten

16469

= diese Konfiguration ist voreingestellt

Ausgänge	mögliche Betriebsart		einstellen mit FB	FB-Eingang	Wert	
					dez	hex
OUT00 ...OUT03	Aus		OUTPUT	MODE	0	00
	Binärer Ausgang	plus-schaltend	OUTPUT	MODE	2	02
	Binärer Ausgang mit Diagnose	plus-schaltend	OUTPUT	MODE	15	0F
	Binärer Ausgang mit Diagnose und Protection	plus-schaltend	OUTPUT	MODE	16	10
	analoger Ausgang mit Pulsweitenmodulation		PWM1000			
	analoger stromgeregelter Ausgang		CURRENT_CONTROL			
OUT04 ...OUT07	Aus		OUTPUT	MODE	0	00
	Binärer Ausgang	plus-schaltend	OUTPUT	MODE	2	02
	Binärer Ausgang	minus-schaltend	OUTPUT	MODE	13	0D
	analoger Ausgang mit Pulsweitenmodulation		PWM1000			
	analoger Ausgang mit Pulsweitenmodulation	minus-regelnd	PWM1000_LOW			
	analoger stromgeregelter Ausgang		CURRENT_CONTROL			
	H-Brücke mit PWM		H_BRIDGE			

Betriebsarten mit folgendem Funktionsbaustein einstellen:

CURRENT_CONTROL (→ Seite 181)	Stromregler für einen PWMi-Ausgangskanal
H_BRIDGE (→ Seite 189)	H-Brücke an einem PWM-Kanalpaar
OUTPUT (→ Seite 183)	weist einem Ausgangskanal eine Betriebsart zu liefert den aktuellen Zustand am gewählten Kanal
PWM1000 (→ Seite 185)	initialisiert und parametriert einen PWM-fähigen Ausgangskanal das Puls-Pausen-Verhältnis kann in 1 ‰-Schritten angegeben werden
PWM1000_LOW (→ Seite 187)	initialisiert und parametriert einen PWM-fähigen Ausgangskanal minus-schaltend das Puls-Pausen-Verhältnis kann in 1 ‰-Schritten angegeben werden

© ifm electronic gmbh www.ifm.com

7.3 Fehler-Tabellen

Inhalt

Fehlermerker 218
Fehler: CAN / CANopen 218

19606

7.3.1 Fehlermerker

19608

→ Kapitel ***Systemmerker*** (→ Seite 214)

7.3.2 Fehler: CAN / CANopen

19610
19604

→ Systemhandbuch "Hintergrundwissen ecomat*mobile*"
→ Kapitel ***CAN / CANopen: Fehler und Fehlerbehandlung***

EMCY-Codes: CANx

13094

Die Angaben für CANx gelten für jede der CAN-Schnittstellen.

EMCY-Code Objekt 0x1003		Objekt 0x1001	herstellerspezifische Informationen					
Byte 0 [hex]	Byte 1 [hex]	Byte 2 [hex]	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Beschreibung
00	80	11	---	---	---	---	---	CANx Monitoring SYNC-Error (nur Slave)
00	81	11	---	---	---	---	---	CANx Warngrenze (> 96)
10	81	11	---	---	---	---	---	CANx Empfangspuffer Überlauf
11	81	11	---	---	---	---	---	CANx Sendepuffer Überlauf
30	81	11	---	---	---	---	---	CANx Guard-/Heartbeat-Error (nur Slave)

© ifm electronic gmbh www.ifm.com

EMCY-Codes: E/As, System

8412

EMCY-Code Objekt 0x1003		Objekt 0x1001	herstellerspezifische Informationen					
Byte 0 [hex]	**Byte 1 [hex]**	**Byte 2 [hex]**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Beschreibung**
00	21	03	I0 LSB	I0 MSB				Leiterbruch Eingänge
08	21	03	I0 LSB	I0 MSB				Kurzschluss Eingänge
10	21	03	I0 LSB	I0 MSB				Überstrom 4…20 mA
00	23	03	Q0 LSB	Q0 MSB				Leiterbruch Ausgänge
08	23	03	Q0 LSB	Q0 MSB				Kurzschluss Ausgänge
00	31	05						Versorgungsspannung VBBS
00	33	05						Klemmenspannung VBBO
08	33	05						Ausgangsspannung VBBR
00	42	09						Übertemperatur

Im CANopen-Stack ist noch keiner dieser EMCY-Codes fix implementiert. Vorschlag:

► Diese EMCY-Codes mit CANOPEN_SENDEMCYMESSAGE erzeugen.

© ifm electronic gmbh

8 Begriffe und Abkürzungen

A

Adresse

Das ist der „Name“ des Teilnehmers im Bus. Alle Teilnehmer benötigen eine unverwechselbare, eindeutige Adresse, damit der Austausch der Signale fehlerfrei funktioniert.

Anleitung

Übergeordnetes Wort für einen der folgenden Begriffe:
Montageanleitung, Datenblatt, Benutzerinformation, Bedienungsanleitung, Gerätehandbuch, Installationsanleitung, Onlinehilfe, Systemhandbuch, Programmierhandbuch, usw.

Anwendungsprogramm

Software, die speziell für die Anwendung vom Hersteller in die Maschine programmiert wird. Die Software enthält üblicherweise logische Sequenzen, Grenzwerte und Ausdrücke zum Steuern der entsprechenden Ein- und Ausgänge, Berechnungen und Entscheidungen.

Architektur

Spezifische Konfiguration von Hardware- und/oder Software-Elementen in einem System.

B

Baud

Baud, Abk.: Bd = Maßeinheit für die Geschwindigkeit bei der Datenübertragung. Baud ist nicht zu verwechseln mit "bits per second" (bps, Bit/s). Baud gibt zwar die Anzahl von Zustandsänderungen (Schritte, Takte) pro Sekunde auf einer Übertragungsstrecke an. Aber es ist nicht festgelegt, wie viele Bits pro Schritt übertragen werden. Der Name Baud geht auf den französischen Erfinder J. M. Baudot zurück, dessen Code für Telexgeräte verwendet wurde.
1 MBd = 1024 x 1024 Bd = 1 048 576 Bd

Bestimmungsgemäße Verwendung

Das ist die Verwendung eines Produkts in Übereinstimmung mit den in der Anleitung bereitgestellten Informationen.

Bootloader

Im Auslieferungszustand enthalten ecomat*mobile*-Controller nur den Bootloader.
Der Bootloader ist ein Startprogramm, mit dem das Laufzeitsystem und das Anwendungsprogramm auf dem Gerät nachgeladen werden können.
Der Bootloader enthält Grundroutinen...
- zur Kommunikation der Hardware-Module untereinander,
- zum Nachladen des Laufzeitsystems.

Der Bootloader ist das erste Software-Modul, das im Gerät gespeichert sein muss.

Bus

Serielle Datenübertragung mehrerer Teilnehmer an derselben Leitung.

© ifm electronic gmbh www.ifm.com

C

CAN

CAN = **C**ontroller **A**rea **N**etwork
CAN gilt als Feldbussystem für größere Datenmengen, das prioritätengesteuert arbeitet. Es gibt mehrere höhere Protokolle, die auf CAN aufsetzen, z. B. 'CANopen' oder 'J1939'.

CAN-Stack

CAN-Stack = Software-Komponente, die sich um die Verarbeitung von CAN-Telegramme kümmert.

CiA

CiA = CAN in Automation e.V.
Anwender- und Herstellerorganisation in Erlangen, Deutschland. Definitions- und Kontrollorgan für das CANopen-Protokoll.
Homepage → www.can-cia.org

CiA DS 304

DS = **D**raft **S**tandard
CANopen-Geräteprofil für sichere Kommunikation

CiA DS 401

DS = **D**raft **S**tandard
CANopen-Geräteprofil für digitale und analoge E/A-Baugruppen

CiA DS 402

DS = **D**raft **S**tandard
CANopen-Geräteprofil für Antriebe

CiA DS 403

DS = **D**raft **S**tandard
CANopen-Geräteprofil für Bediengeräte

CiA DS 404

DS = **D**raft **S**tandard
CANopen-Geräteprofil für Messtechnik und Regler

CiA DS 405

DS = **D**raft **S**tandard
CANopen-Spezifikation der Schnittstelle zu programmierbaren Steuerungen (IEC 61131-3)

CiA DS 406

DS = **D**raft **S**tandard
CANopen-Geräteprofil für Drehgeber / Encoder

CiA DS 407

DS = **D**raft **S**tandard
CANopen-Anwendungsprofil für den öffentlichen Nahverkehr

©ifm electronic gmbh www.ifm.com

COB-ID

COB = **C**ommunication **Ob**ject = Kommunikationsobjekt
ID = **Id**entifier = Kennung
ID eines CANopen-Kommunikationsobjekts
Entspricht dem Identifier der CAN-Nachricht, mit der das Kommunikationsobjekt über den CAN-Bus gesendet wird.

CODESYS

CODESYS® ist eingetragene Marke der 3S – Smart Software Solutions GmbH, Deutschland.
'CODESYS for Automation Alliance℠' vereinigt Firmen der Automatisierungsindustrie, deren Hardware-Geräte alle mit dem weit verbreiteten IEC 61131-3 Entwicklungswerkzeug CODESYS® programmiert werden.
Homepage → www.codesys.com

CSV-Datei

CSV = **C**omma **S**eparated **V**alues (auch: **C**haracter **S**eparated **V**alues)
Eine CSV-Datei ist eine Textdatei zur Speicherung oder zum Austausch einfach strukturierter Daten.
Die Dateinamen-Erweiterung lautet `.csv`.

Beispiel: Quell-Tabelle mit Zahlenwerten:

Wert 1.0	Wert 1.1	Wert 1.2	Wert 1.3
Wert 2.0	Wert 2.1	Wert 2.2	Wert 2.3
Wert 3.0	Wert 3.1	Wert 3.2	Wert 3.3

Daraus entsteht folgende CSV-Datei:

```
Wert 1.0;Wert 1.1;Wert 1.2;Wert 1.3
Wert 2.0;Wert 2.1;Wert 2.2;Wert 2.3
Wert 3.0;Wert 3.1;Wert 3.2;Wert 3.3
```

D

Datentyp

Abhängig vom Datentyp können unterschiedlich große Werte gespeichert werden.

Datentyp	min. Wert	max. Wert	Größe im Speicher
BOOL	FALSE	TRUE	8 Bit = 1 Byte
BYTE	0	255	8 Bit = 1 Byte
WORD	0	65 535	16 Bit = 2 Bytes
DWORD	0	4 294 967 295	32 Bit = 4 Bytes
SINT	-128	127	8 Bit = 1 Byte
USINT	0	255	8 Bit = 1 Byte
INT	-32 768	32 767	16 Bit = 2 Bytes
UINT	0	65 535	16 Bit = 2 Bytes
DINT	-2 147 483 648	2 147 483 647	32 Bit = 4 Bytes
UDINT	0	4 294 967 295	32 Bit = 4 Bytes
REAL	$-3{,}402823466 \cdot 10^{38}$	$3{,}402823466 \cdot 10^{38}$	32 Bit = 4 Bytes
ULINT	0	18 446 744 073 709 551 615	64 Bit = 8 Bytes
STRING			number of char. + 1

© ifm electronic gmbh www.ifm.com

DC

Direct **C**urrent = Gleichstrom

Diagnose

Bei der Diagnose wird der "Gesundheitszustand" des Gerätes geprüft. Es soll festgestellt werden, ob und gegebenenfalls welche →Fehler im Gerät vorhanden sind.

Je nach Gerät können auch die Ein- und Ausgänge auf einwandfreie Funktion überwacht werden:
- Drahtbruch,
- Kurzschluss,
- Wert außerhalb des Sollbereichs.

Zur Diagnose können Konfigurations-Dateien herangezogen werden, die während des "normalen" Betriebs des Gerätes erzeugt wurden.
Der korrekte Start der Systemkomponenten wird während der Initialisierungs- und Startphase überwacht.
Zur weiteren Diagnose können auch Selbsttests durchgeführt werden.

Dither

to dither (engl.) = schwanken / zittern.
Dither ist ein Bestandteil der →PWM-Signale zum Ansteuern von Hydraulik-Ventilen. Für die elektromagnetischen Antriebe von Hydraulik-Ventilen hat sich herausgestellt, dass sich die Ventile viel besser regeln lassen, wenn das Steuersignal (PWM-Impulse) mit einer bestimmten Frequenz der PWM-Frequenz überlagert wird. Diese Dither-Frequenz muss ein ganzzahliger Teil der PWM-Frequenz sein.

DLC

Data **L**ength **C**ode = bei CANopen die Anzahl der Daten-Bytes in einer Nachricht.
Für →SDO: DLC = 8

DRAM

DRAM = **D**ynamic **R**andom **A**ccess **M**emory.
Technologie für einen elektronischen Speicherbaustein mit wahlfreiem Zugriff (Random Access Memory, RAM). Das speichernde Element ist dabei ein Kondensator, der entweder geladen oder entladen ist. Über einen Schalttransistor wird er zugänglich und entweder ausgelesen oder mit neuem Inhalt beschrieben. Der Speicherinhalt ist flüchtig: die gespeicherte Information geht bei fehlender Betriebsspannung oder zu später Wiederauffrischung verloren.

DTC

DTC = **D**iagnostic **T**rouble **C**ode = Fehler-Code
Beim Protokoll J1939 werden Störungen und Fehler über zugeordnete Nummern – den DTCs – verwaltet und gemeldet.

E

ECU

(1) **E**lectronic **C**ontrol **U**nit = Steuergerät oder Mikrocontroller
(2) **E**ngine **C**ontrol **U**nit = Steuergerät eines Motors

© ifm electronic gmbh www.ifm.com

EDS-Datei

EDS = **E**lectronic **D**ata **S**heet = elektronisch hinterlegtes Datenblatt, z.B. für:
- Datei für das Objektverzeichnis im CANopen-Master,
- CANopen-Gerätebeschreibungen.

Via EDS können vereinfacht Geräte und Programme ihre Spezifikationen austauschen und gegenseitig berücksichtigen.

Embedded Software

System-Software, Grundprogramm im Gerät, praktisch das →Laufzeitsystem.
Die Firmware stellt die Verbindung her zwischen der Hardware des Gerätes und dem Anwendungsprogramm. Die Firmware wird vom Hersteller der Steuerung als Teil des Systems geliefert und kann vom Anwender nicht verändert werden.

EMCY

Abkürzung für Emergency (engl.) = Notfall
Nachricht im CANopen-Protokoll, mit der Fehler gemeldet werden.

EMV

EMV = **E**lektro-**M**agnetische **V**erträglichkeit.
Gemäß der EG-Richtlinie (2004/108/EG) zur elektromagnetischen Verträglichkeit (kurz EMV-Richtlinie) werden Anforderungen an die Fähigkeit von elektrischen und elektronischen Apparaten, Anlagen, Systemen oder Bauteilen gestellt, in der vorhandenen elektromagnetischen Umwelt zufriedenstellend zu arbeiten. Die Geräte dürfen ihre Umgebung nicht stören und dürfen sich von äußerlichen elektromagnetischen Störungen nicht ungünstig beeinflussen lassen.

Ethernet

Ethernet ist eine weit verbreitete, herstellerneutrale Netzwerktechnologie, mit der Daten mit einer Geschwindigkeit von 10 bis 10 000 Millionen Bit pro Sekunde (Mbps) übertragen werden können. Ethernet gehört zu der Familie der sogenannten „bestmöglichen Datenübermittlung“ auf einem nicht exklusiven Übertragungsmedium. 1972 entwickelt, wurde das Konzept 1985 als IEEE 802.3 spezifiziert.

EUC

EUC = **E**quipment **U**nder **C**ontrol (kontrollierte Einrichtung).
EUC ist eine Einrichtung, Maschine, Gerät oder Anlage, verwendet zur Fertigung, Stoffumformung, zum Transport, zu medizinischen oder anderen Tätigkeiten (→ IEC 61508-4, Abschnitt 3.2.3). Das EUC umfasst also alle Einrichtungen, Maschinen, Geräte oder Anlagen, die →Gefährdungen verursachen können und für die sicherheitsgerichtete Systeme erforderlich sind.
Falls eine vernünftigerweise vorhersehbare Aktivität oder Inaktivität zu durch das EUC verursachten Gefährdungen mit unvertretbarem Risiko führt, sind Sicherheitsfunktionen erforderlich, um einen sicheren Zustand für das EUC zu erreichen oder aufrecht zu erhalten. Diese Sicherheitsfunktionen werden durch ein oder mehrere sicherheitsgerichtete Systeme ausgeführt.

F

Fehlanwendung

Das ist die Verwendung eines Produkts in einer Weise, die vom Konstrukteur nicht vorgesehen ist.
Eine Fehlanwendung führt meist zu einer →Gefährdung von Personen oder Sachen.
Vor vernünftigerweise, vorhersehbaren Fehlanwendungen muss der Hersteller des Produkts in seinen Benutzerinformationen warnen.

© ifm electronic gmbh www.ifm.com

FiFo

FIFO (**F**irst **I**n, **F**irst **O**ut) = Arbeitsweise des Stapelspeichers: Das Datenpaket, das zuerst in den Stapelspeicher geschrieben wurde, wird auch als erstes gelesen. Pro Identifier steht ein solcher Zwischenspeicher (als Warteschlange) zur Verfügung.

Flash-Speicher

Flash-ROM (oder Flash-EPROM oder Flash-Memory) kombiniert die Vorteile von Halbleiterspeicher und Festplatten. Die Daten werden allerdings wie bei einer Festplatte blockweise in Datenblöcken zu 64, 128, 256, 1024, ... Byte zugleich geschrieben und gelöscht.

Vorteile von Flash-Speicher

- Die gespeicherten Daten bleiben auch bei fehlender Versorgungsspannung erhalten.
- Wegen fehlender beweglicher Teile ist Flash geräuschlos, unempfindlich gegen Erschütterungen und magnetische Felder.

Nachteile von Flash-Speicher

- Begrenzte Zahl von Schreib- bzw. Löschvorgängen, die eine Speicherzelle vertragen kann:
  - Multi-Level-Cells: typ. 10 000 Zyklen
  - Single-Level-Cells: typ. 100 000 Zyklen
- Da ein Schreibvorgang Speicherblöcke zwischen 16 und 128 kByte gleichzeitig beschreibt, werden auch Speicherzellen beansprucht, die gar keiner Veränderung bedürfen.

FRAM

FRAM, oder auch FeRAM, bedeutet **Fe**rroelectric **R**andom **A**ccess **M**emory. Der Speicher- und Löschvorgang erfolgt durch eine Polarisationsänderung in einer ferroelektrischen Schicht.
Vorteile von FRAM gegenüber herkömmlichen Festwertspeichern:

- nicht flüchtig,
- kompatibel zu gängigen EEPROMs, jedoch:
- Zugriffszeit ca. 100 ns,
- fast unbegrenzt viele Zugriffszyklen möglich.

H

Heartbeat

Heartbeat (engl.) = Herzschlag.
Die Teilnehmer senden regelmäßig kurze Signale. So können die anderen Teilnehmer prüfen, ob ein Teilnehmer ausgefallen ist.

HMI

HMI = **H**uman **M**achine **I**nterface = Mensch-Maschine-Schnittstelle

I

ID – Identifier

ID = **Id**entifier = Kennung

Name zur Unterscheidung der an einem System angeschlossenen Geräte / Teilnehmer oder der zwischen den Teilnehmern ausgetauschten Nachrichtenpakete.

© ifm electronic gmbh www.ifm.com

IEC 61131

Norm: Grundlagen Speicherprogrammierbarer Steuerungen
- Teil 1: Allgemeine Informationen
- Teil 2: Betriebsmittelanforderungen und Prüfungen
- Teil 3: Programmiersprachen
- Teil 5: Kommunikation
- Teil 7: Fuzzy-Control-Programmierung

IEC-User-Zyklus

IEC-User-Zyklus = SPS-Zyklus im CODESYS-Anwendungsprogramm.

IP-Adresse

IP = **I**nternet **P**rotocol = Internet-Protokoll.
Die IP-Adresse ist eine Nummer, die zur eindeutigen Identifizierung eines Internet-Teilnehmers notwendig ist. Zur besseren Übersicht wird die Nummer in 4 dezimalen Werten geschrieben, z. B. 127.215.205.156.

ISO 11898

Norm: Straßenfahrzeuge – CAN-Protokoll
- Teil 1: Bit-Übertragungsschicht und physikalische Zeichenabgabe
- Teil 2: High-speed medium access unit
- Teil 3: Fehlertolerante Schnittstelle für niedrige Geschwindigkeiten
- Teil 4: Zeitgesteuerte Kommunikation
- Teil 5: High-speed medium access unit with low-power mode

ISO 11992

Norm: Straßenfahrzeuge – Austausch von digitalen Informationen über elektrische Verbindungen zwischen Zugfahrzeugen und Anhängefahrzeugen
- Teil 1: Bit-Übertragungsschicht und Sicherungsschicht
- Teil 2: Anwendungsschicht für die Bremsausrüstung
- Teil 3: Anwendungsschicht für andere als die Bremsausrüstung
- Teil 4: Diagnose

ISO 16845

Norm: Straßenfahrzeuge – Steuergerätenetz (CAN) – Prüfplan zu Konformität

J

J1939

→ SAE J1939

K

Klemme 15

Klemme 15 ist in Fahrzeugen die vom Zündschloss geschaltete Plusleitung.

© ifm electronic gmbh www.ifm.com

L

Laufzeitsystem

Grundprogramm im Gerät, stellt die Verbindung her zwischen der Hardware des Gerätes und dem Anwendungsprogramm.
→ Kapitel *Software-Module für das Gerät* (→ Seite 27)

LED

LED = **L**ight **E**mitting **D**iode = Licht aussendende Diode.
Leuchtdiode, auch Luminiszenzdiode, ein elektronisches Element mit hoher, farbiger Leuchtkraft auf kleinem Volumen bei vernachlässigbarer Verlustleistung.

Link

Ein Link ist ein Querverweis zu einer anderen Stelle im Dokument oder auf ein externes Dokument.

LSB

Least **S**ignificant **B**it/Byte = Niederwertigstes Bit/Byte in einer Reihe von Bit/Bytes.

M

MAC-ID

MAC = **M**anufacturer‘s **A**ddress **C**ode
= Hersteller-Seriennummer.
→ID = **Id**entifier = Kennung
Jede Netzwerkkarte verfügt über eine so genannte MAC-Adresse, ein unverwechselbarer, auf der ganzen Welt einzigartiger Zahlencode – quasi eine Art Seriennummer. So eine MAC-Adresse ist eine Aneinanderreihung von 6 Hexadezimalzahlen, etwa "00-0C-6E-D0-02-3F".

Master

Wickelt die komplette Organisation auf dem →Bus ab. Der Master entscheidet über den zeitlichen Buszugriff und fragt die →Slaves zyklisch ab.

MMI

MMI = **M**ensch-**M**aschine-**I**nterface
→ *HMI* (→ Seite 225)

MRAM

MRAM = **M**agnetoresistive **R**andom **A**ccess **M**emory
Die Informationen werden mit magnetischen Ladungselementen gespeichert. Dabei wird die Eigenschaft bestimmter Materialien ausgenutzt, die ihren elektrischen Widerstand unter dem Einfluss magnetischer Felder ändern.
Vorteile von MRAM gegenüber herkömmlichen Festwertspeichern:
- nicht flüchtig (wie FRAM), jedoch:
- Zugriffszeit nur ca. 35 ns,
- unbegrenzt viele Zugriffszyklen möglich.

MSB

Most **S**ignificant **B**it/Byte = Höchstwertiges Bit/Byte einer Reihe von Bits/Bytes.

© ifm electronic gmbh www.ifm.com

N

NMT

NMT = **N**etwork **M**anagemen**t** = Netzwerk-Verwaltung (hier: im CANopen-Protokoll).
Der NMT-Master steuert die Betriebszustände der NMT-Slaves.

Node

Node (engl.) = Knoten. Damit ist ein Teilnehmer im Netzwerk gemeint.

Node Guarding

Node (engl.) = Knoten, hier: Netzwerkteilnehmer
Guarding (engl.) = Schutz
Parametrierbare, zyklische Überwachung von jedem entsprechend konfigurierten →Slave. Der →Master prüft, ob die Slaves rechtzeitig antworten. Die Slaves prüfen, ob der Master regelmäßig anfragt. Somit können ausgefallene Netzwerkteilnehmer schnell erkannt und gemeldet werden.

O

Obj / Objekt

Oberbegriff für austauschbare Daten / Botschaften innerhalb des CANopen-Netzwerks.

Objektverzeichnis

Das **Ob**jekt**v**erzeichnis OBV enthält alle CANopen-Kommunikationsparameter eines Gerätes, sowie gerätespezifische Parameter und Daten.

OBV

Das **Ob**jekt**v**erzeichnis OBV enthält alle CANopen-Kommunikationsparameter eines Gerätes, sowie gerätespezifische Parameter und Daten.

OPC

OPC = **O**LE for **P**rocess **C**ontrol = Objektverknüpfung und -einbettung für Prozesssteuerung
Standardisierte Software-Schnittstelle zur herstellerunabhängigen Kommunikation in der Automatisierungstechnik
OPC-Client (z.B. Gerät zum Parametrieren oder Programmieren) meldet sich nach dem Anschließen am OPC-Server (z.B. Automatisierungsgerät) automatisch bei diesem an und kommuniziert mit ihm.

operational

Operational (engl.) = betriebsbereit
Betriebszustand eines CANopen-Teilnehmers. In diesem Modus können →SDOs, →NMT-Kommandos und →PDOs übertragen werden.

P

PC-Karte

→ PCMCIA-Karte

© ifm electronic gmbh www.ifm.com

PCMCIA-Karte

PCMCIA = Personal Computer Memory Card International Association, ein Standard für Erweiterungskarten mobiler Computer.
Seit der Einführung des Cardbus-Standards 1995 werden PCMCIA-Karten auch als PC-Karte (engl.: PC Card) bezeichnet.

PDM

PDM = **P**rocess and **D**ialog **M**odule = **P**rozess- und **D**ialog-**M**onitor.
Gerät zur Kommunikation des Bedieners mit der Maschine / Anlage.

PDO

PDO = **P**rocess **D**ata **O**bject = Nachrichten-Objekt mit Prozessdaten.
Die zeitkritischen Prozessdaten werden mit Hilfe der "Process Data Objects" (PDOs) übertragen. Die PDOs können beliebig zwischen den einzelnen Knoten ausgetauscht werden (PDO-Linking). Zusätzlich wird festgelegt, ob der Datenaustausch ereignisgesteuert (asynchron) oder synchronisiert erfolgen soll. Je nach der Art der zu übertragenden Daten kann die richtige Wahl der Übertragungsart zu einer erheblichen Entlastung des →CAN-Bus führen.
Dem Protokoll entsprechend, sind diese Dienste nicht bestätigte Dienste: es gibt keine Kontrolle, ob die Nachricht auch beim Empfänger ankommt. Netzwerkvariablen-Austausch entspricht einer "1-zu-n-Verbindung" (1 Sender zu n Empfängern).

PDU

PDU = **P**rotocol **D**ata **U**nit = Protokoll-Daten-Einheit.
Die PDU ist ein Begriff aus dem →CAN-Protokoll →SAE J1939. Sie bezeichnet einen Bestandteil der Ziel- oder Quelladresse.

PES

Programable **e**lectronic **s**ystem = Programmierbares elektronisches System ...
- zur Steuerung, zum Schutz oder zur Überwachung,
- auf der Basis einer oder mehrerer programmierbarer Geräte,
- einschließlich aller Elemente dieses Systems, wie Ein- und Ausgabegeräte.

PGN

PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer
PGN = PDU Format (PF) + PDU Source (PS)
Die Parameter-Gruppennummer ist ein Begriff aus dem →CAN-Protokoll →SAE J1939. Sie fasst die Teiladressen PF und PS zusammen.

PID-Regler

Der PID-Regler (proportional–integral–derivative controller) besteht aus folgenden Anteilen:
- P = Proportional-Anteil
- I = Integral-Anteil
- D = Differential-Anteil (jedoch nicht beim Controller CR04nn, CR253n).

Piktogramm

Piktogramme sind bildhafte Symbole, die eine Information durch vereinfachte grafische Darstellung vermitteln (→ Kapitel ***Was bedeuten die Symbole und Formatierungen?*** (→ Seite 7)).

© ifm electronic gmbh www.ifm.com

Pre-Op

Pre-Op = PRE-OPERATIONAL mode (engl.) = Zustand vor 'betriebsbereit'.
Betriebszustand eines CANopen-Teilnehmers. Nach dem Einschalten der Versorgungsspannung geht jeder Teilnehmer automatisch in diesem Zustand. Im CANopen-Netz können in diesem Modus nur →SDOs und →NMT-Kommandos übertragen werden, jedoch keine Prozessdaten.

Prozessabbild

Mit Prozessabbild bezeichnet man den Zustand der Ein- und Ausgänge, mit denen die SPS innerhalb eines →Zyklusses arbeitet.

- Am Zyklus-Beginn liest die SPS die Zustände aller Eingänge in das Prozessabbild ein. Während des Zyklusses kann die SPS Änderungen an den Eingängen nicht erkennen.
- Im Laufe des Zyklusses werden die Ausgänge nur virtuell (im Prozessabbild) geändert.
- Am Zyklus-Ende schreibt die SPS die virtuellen Ausgangszustände auf die realen Ausgänge.

PWM

PWM = Puls-Weiten-Modulation
Bei dem PWM-Ausgangssignal handelt es sich um ein getaktetes Signal zwischen GND und Versorgungsspannung.
Innerhalb einer festen Periode (PWM-Frequenz) wird das Puls-/Pausenverhältnis variiert. Durch die angeschlossene Last stellt sich je nach Puls-/Pausenverhältnis der entsprechende Effektivstrom ein.

R

ratiometrisch

Ratio (lat.) = Verhältnis
Messungen können auch ratiometrisch erfolgen = Verhältnismessung. Wenn das Ausgangssinal eines Sensors proportional zu seiner Versorgungsspannung ist, kann durch ratiometrische Messung (= Messung im Verhältnis zur Versorgung) der Einfluss von Schwankungen der Versorgung reduziert, im Idealfall sogar beseitigt werden.
→ Analogeingang

RAW-CAN

RAW-CAN bezeichnet das reine →CAN-Protokoll, das ohne ein zusätzliches Kommunikationsprotokoll auf dem CAN-Bus (auf ISO/OSI-Schicht 2) arbeitet. Das CAN-Protokoll ist international nach →ISO 11898-1 definiert und garantiert zusätzlich in →ISO 16845 die Austauschbarkeit von CAN-Chips.

remanent

Remanente Daten sind gegen Datenverlust bei Spannungsausfall geschützt.
Z.B. kopiert das →Laufzeitsystem die remanenten Daten automatisch in einen →Flash-Speicher, sobald die Spannungsversorgung unter einen kritischen Wert sinkt. Bei Wiederkehr der Spannungsversorgung lädt das Laufzeitsystem die remanenten Daten zurück in den Arbeitsspeicher. Dagegen sind die Daten im Arbeitsspeicher einer Steuerung flüchtig und bei Unterbrechung der Spannungsversorgung normalerweise verloren.

ro

ro = read only (engl.) = nur lesen
Unidirektionale Datenübertragung: Daten können nur gelesen werden, jedoch nicht verändert.

© ifm electronic gmbh www.ifm.com

RTC

RTC = **R**eal **T**ime **C**lock = Echtzeituhr
Liefert (batteriegepuffert) aktuell Datum und Uhrzeit. Häufiger Einsatz beim Speichern von Fehlermeldungsprotokollen.

rw

rw = read/write (engl.) = lesen und schreiben
Bidirektionale Datenübertragung: Daten können sowohl gelesen als auch verändert werden.

S

SAE J1939

Das Netzwerkprotokoll SAE J1939 beschreibt die Kommunikation auf einem →CAN-Bus in Nutzfahrzeugen zur Übermittlung von Diagnosedaten (z.B.Motordrehzahl, Temperatur) und Steuerungsinformationen.
Norm: Recommended Practice for a Serial Control and Communications Vehicle Network
- Teil 2: Agricultural and Forestry Off-Road Machinery Control and Communication Network
- Teil 3: On Board Diagnostics Implementation Guide
- Teil 5: Marine Stern Drive and Inboard Spark-Ignition Engine On-Board Diagnostics Implementation Guide
- Teil 11: Physical Layer – 250 kBits/s, Shielded Twisted Pair
- Teil 13: Off-Board Diagnostic Connector
- Teil 15: Reduced Physical Layer, 250 kBits/s, Un-Shielded Twisted Pair (UTP)
- Teil 21: Data Link Layer
- Teil 31: Network Layer
- Teil 71: Vehicle Application Layer
- Teil 73: Application Layer – Diagnostics
- Teil 81: Network Management Protocol

SD-Card

Eine SD Memory Card (Kurzform für **S**ecure **D**igital Memory Card; deutsch: Sichere digitale Speicherkarte) ist ein digitales Speichermedium, das nach dem Prinzip der →Flash-Speicherung arbeitet.

SDO

SDO = **S**ervice **D**ata **O**bject = Nachrichten-Objekt mit Servicedaten.
Das SDO dient dem Zugriff auf Objekte in einem CANopen-Objektverzeichnis. Dabei fordern 'Clients' die gewünschten Daten von 'Servern' an. Die SDOs bestehen immer aus 8 Bytes.
Beispiele:
- Automatische Konfiguration aller →Slaves über SDOs beim Systemstart.
- Auslesen der Fehlernachrichten aus dem →Objektverzeichnis.

Jedes SDO wird auf Antwort überwacht und wiederholt, wenn sich innerhalb der Überwachungszeit der Slave nicht meldet.

Selbsttest

Testprogramm, das aktiv Komponenten oder Geräte testet. Das Programm wird durch den Anwender gestartet und dauert eine gewisse Zeit. Das Ergebnis davon ist ein Testprotokoll (Log-Datei), aus dem entnommen werden kann, was getestet wurde und ob das Ergebnis positiv oder negativ ist.

© ifm electronic gmbh www.ifm.com

Slave

Passiver Teilnehmer am Bus, antwortet nur auf Anfrage des →Masters. Slaves haben im Bus eine eindeutige →Adresse.

Steuerungskonfiguration

Bestandteil der CODESYS-Bedienoberfläche.

► Programmierer teilt dem Programmiersystem mit, welche Hardware programmiert werden soll.
> CODESYS lädt die zugehörigen Bibliotheken.
> Lesen und schreiben der Peripherie-Zustände (Ein-/Ausgänge) ist möglich.

stopped

stopped (engl.) = angehalten
Betriebszustand eines CANopen-Teilnehmers. In diesem Modus werden nur →NMT-Kommandos übertragen.

Symbole

Piktogramme sind bildhafte Symbole, die eine Information durch vereinfachte grafische Darstellung vermitteln (→ Kapitel ***Was bedeuten die Symbole und Formatierungen?*** (→ Seite 7)).

Systemvariable

Variable, auf die via IEC-Adresse oder Symbolname aus der SPS zugegriffen werden kann.

T

Target

Das Target enthält für CODESYS die Hardware-Beschreibung des Zielgeräts, z.B.: Ein- und Ausgänge, Speicher, Dateiablageorte.
Entspricht einem elektronischen Datenblatt.

TCP

Das **T**ransmission **C**ontrol **P**rotocol ist Teil der Protokollfamilie TCP/IP. Jede TCP/IP-Datenverbindung hat einen Sender und einen Empfänger. Dieses Prinzip ist eine verbindungsorientierte Datenübertragung. In der TCP/IP-Protokollfamilie übernimmt TCP als verbindungsorientiertes Protokoll die Aufgabe der Datensicherheit, der Datenflusssteuerung und ergreift Maßnahmen bei einem Datenverlust. (vgl.: →UDP)

Template

Template (englisch = Schablone) ist eine Vorlage, die mit Inhalten gefüllt werden kann.
Hier: Eine Struktur von vorkonfigurierten Software-Elementen als Basis für ein Anwendungsprogramm.

© ifm electronic gmbh www.ifm.com

U

UDP

UDP (**U**ser **D**atagram **P**rotocol) ist ein minimales, verbindungsloses Netzprotokoll, das zur Transportschicht der Internetprotokollfamilie gehört. Aufgabe von UDP ist es, Daten, die über das Internet übertragen werden, der richtigen Anwendung zukommen zu lassen.
Derzeit sind Netzwerkvariablen auf Basis von →CAN und UDP implementiert. Die Variablenwerte werden dabei auf der Basis von Broadcast-Nachrichten automatisch ausgetauscht. In UDP sind diese als Broadcast-Telegramme realisiert, in CAN als →PDOs.
Dem Protokoll entsprechend, sind diese Dienste nicht bestätigte Dienste: es gibt keine Kontrolle, ob die Nachricht auch beim Empfänger ankommt. Netzwerkvariablen-Austausch entspricht einer "1-zu-n-Verbindung" (1 Sender zu n Empfängern).

V

Verwendung, bestimmungsgemäß

Das ist die Verwendung eines Produkts in Übereinstimmung mit den in der Anleitung bereitgestellten Informationen.

W

Watchdog

Der Begriff Watchdog (englisch; Wachhund) wird verallgemeinert für eine Komponente eines Systems verwendet, die die Funktion anderer Komponenten beobachtet. Wird dabei eine mögliche Fehlfunktionen erkannt, so wird dies entweder signalisiert oder geeignete Programm-Verzweigungen eingeleitet. Das Signal oder die Verzweigungen dienen als Auslöser für andere kooperierende Systemkomponenten, die das Problem lösen sollen.

wo

wo = write only (engl.) = nur schreiben
Unidirektionale Datenübertragung: Daten können nur verändert werden, jedoch nicht gelesen.

Z

Zykluszeit

Das ist die Zeit für einen Zyklus. Das SPS-Programm läuft einmal komplett durch.

Je nach ereignisgesteuerten Verzweigungen im Programm kann dies unterschiedlich lange dauern.

© ifm electronic gmbH www.ifm.com

9 Index

A

Adressbelegung ... 215
Adressbelegung der Eingänge ... 215
Adressbelegung Ein-/Ausgänge ... 215
Adressbelegung und E/A-Betriebsarten ... 214
Adresse ... 220
Analogeingänge
  Konfiguration und Diagnose ... 50
Analog-Eingänge ... 14
Angaben zum Gerät ... 12
Anhang ... 214
Anlaufverhalten der Steuerung ... 11
Anleitung ... 220
Anschlussbelegung ... 24
Anwendungsprogramm ... 28, 220
Anwendungsprogramm erstellen ... 32
Architektur ... 220
Ausgänge
  Adressbelegung ... 215
  Betriebsarten ... 217
Ausgänge (Technologie) ... 18
Ausgänge konfigurieren ... 52
Ausgangsgruppe Q0 (OUT0...OUT3) ... 20
Ausgangsgruppe Q1 (OUT4...OUT7) ... 22

B

Baud ... 220
Baustein-Ausgänge ... 66
Bausteine
  Ausgangsfunktionen ... 180
  CANopen ... 93
  CANopen Emergency ... 132
  CANopen Guarding ... 128
  CANopen Netzwerkmanagement ... 102
  CANopen Objektverzeichnis ... 106
  CANopen SDOs ... 111
  CANopen Status ... 93
  CANopen SYNC ... 124
  Eingangswerte verarbeiten ... 170
  RAW-CAN (Layer 2) ... 67
  RAW-CAN Daten empfangen ... 73
  RAW-CAN Daten senden ... 83
  RAW-CAN Remote ... 89
  RAW-CAN Status ... 67
  SAE J1939 ... 138
  SAE J1939 Diagnose ... 162
  SAE J1939 Empfangen ... 149
  SAE J1939 Request ... 146
  SAE J1939 Senden ... 154
  SAE J1939 Status ... 138
  System ... 194
Beachten! ... 10
Beispielablauf für Reaktion auf Fehlermeldungen ... 213
Bestimmungsgemäße Verwendung ... 220
Betriebsstundenzähler ... 205
Betriebszustände ... 34
Bibliothek ifm_CANopen_NT_Vxxyyzz.LIB ... 62
Bibliothek ifm_CR0411_util_Vxxyyzz.LIB ... 64
Bibliothek ifm_CR0411_V03yyzz.LIB ... 60
Bibliothek ifm_J1939_NT_Vxxyyzz.LIB ... 63
Bibliothek ifm_RAWCan_NT_Vxxyyzz.LIB ... 61
Bibliotheken ... 29
Binärausgänge
  Konfiguration und Diagnose ... 53
Binäreingänge
  Konfiguration und Diagnose ... 50
Binär-Eingänge ... 15
Bootloader ... 28, 220
Bus ... 220

C

CAN ... 221
  Schnittstellen und Protokolle ... 26
CAN / CANopen
  Fehler und Fehlerbehandlung ... 213
CAN_ENABLE ... 68
CAN_RECOVER ... 69
CAN_REMOTE_REQUEST ... 90
CAN_REMOTE_RESPONSE ... 91
CAN_RX ... 74
CAN_RX_ENH ... 75
CAN_RX_ENH_FIFO ... 77
CAN_RX_RANGE ... 79
CAN_RX_RANGE_FIFO ... 81
CAN_SETDOWNLOADID ... 70
CAN_STATUS ... 71
CAN_TX ... 84
CAN_TX_ENH ... 85
CAN_TX_ENH_CYCLIC ... 87
CANOPEN_ENABLE ... 94
CANOPEN_GETBUFFERFLAGS ... 96
CANOPEN_GETEMCYMESSAGES ... 133
CANOPEN_GETERRORREGISTER ... 135
CANOPEN_GETGUARDHBERRLIST ... 129
CANOPEN_GETGUARDHBSTATSLV ... 130
CANOPEN_GETNMTSTATESLAVE ... 103
CANOPEN_GETODCHANGEDFLAG ... 107
CANOPEN_GETSTATE ... 98
CANOPEN_GETSYNCSTATE ... 125
CANOPEN_NMTSERVICES ... 104
CANOPEN_READOBJECTDICT ... 108
CANOPEN_SDOREAD ... 112
CANOPEN_SDOREADBLOCK ... 114
CANOPEN_SDOREADMULTI ... 116
CANOPEN_SDOWRITE ... 118
CANOPEN_SDOWRITEBLOCK ... 120
CANOPEN_SDOWRITEMULTI ... 122
CANOPEN_SENDEMCYMESSAGE ... 136
CANOPEN_SETSTATE ... 100
CANOPEN_SETSYNCSTATE ... 127
CANOPEN_WRITEOBJECTDICT ... 109
CAN-Schnittstellen ... 26
CAN-Schnittstellen deklarieren (z.B. CR1080) ... 45
CAN-Stack ... 221
CiA ... 221
CiA DS 304 ... 221
CiA DS 401 ... 221
CiA DS 402 ... 221
CiA DS 403 ... 221
CiA DS 404 ... 221
CiA DS 405 ... 221
CiA DS 406 ... 221
CiA DS 407 ... 221
COB-ID ... 222
CODESYS ... 222

© ifm electronic gmbh www.ifm.com

CODESYS-Programmierhandbuch 6
Copyright 5
CSV-Datei 222
CURRENT_CONTROL 181

D

Datentyp 222
DC 223
Definition
  Kurzschluss 18
  Überlast 18
Diagnose 212, 223
  binäre Ausgänge (via Strom- und Spannungsmessung) 20, 22
  Kurzschluss 21, 23
  Leiterbruch (via Spannungsmessung) 21, 23
  Leiterbruch (via Strommessung) 20, 22
  Überlast 20, 22
Diagnose der Ausgänge konfigurieren 53
Diagnose der Eingänge aktivieren 51
Diagnose und Fehlerbehandlung 212
Dither 223
DLC 223
DRAM 223
DTC 223

E

ECU 223
EDS-Datei 224
Eigenschutz des Ausgangs 19
Eingänge
  Adressbelegung 215
  Betriebsarten 216
Eingänge (Technologie) 14
Eingänge konfigurieren 48
Eingangsgruppe I0 (IN0...IN3) 16
Eingangsgruppe I1 (IN4...IN7) 16
Eingangswerte verarbeiten 170
Einsatz als Binäreingänge 51
Embedded Software 224
EMCY 224
EMCY-Codes
  CANx 218
  E/As, System 219
EMV 224
ERROR-Zustand 35
Ethernet 224
EUC 224

F

FASTCOUNT 171
FATAL-ERROR-Zustand 36
FB, FUN, PRG in CODESYS 30
FBs für PWM-Funktionen 55
Fehlanwendung 224
Fehler 212
  CAN / CANopen 218
Fehlermerker 218
Fehler-Tabellen 218
FiFo 225
FLASH_INFO 195
FLASH_READ 196
Flash-Speicher 225
FLASH-Speicher 13
FRAM 13, 225
Funktionskonfiguration 47
Funktionskonfiguration der Ein- und Ausgänge 47
Funktionskonfiguration, allgemein 46

G

Gerätekonfiguration 39
GET_APP_INFO 197
GET_HW_INFO 198
GET_IDENTITY 199
GET_SW_INFO 200
GET_SW_VERSION 201
Grenzen für CAN in diesem Gerät 38
Grenzen für CAN J1939 in diesem Gerät 38
Grenzen für CANopen in diesem Gerät 38

H

H_BRIDGE 189
Hardware-Aufbau 13
Hardware-Beschreibung 12
Heartbeat 225
Hinweise zur Anschlussbelegung 24
Historie der Anleitung (CR041n) 9
HMI 225

I

ID – Identifier 225
IEC 61131 226
IEC-User-Zyklus 226
ifm weltweit • ifm worldwide • ifm à l'échelle internationale 243
ifm-Bausteine für das Gerät CR0411 65
ifm-Bibliotheken für das Gerät CR0411 59
ifm-Funktionselemente 59
ifm-Maintenance-Tool nutzen 33
INC_ENCODER 173
INIT-Zustand (Reset) 34
INPUT 176
Installation verifizieren 41
IP-Adresse 226
ISO 11898 226
ISO 11992 226
ISO 16845 226

J

J1939 226
J1939_DM1RX 163
J1939_DM1TX 165
J1939_DM1TX_CFG 168
J1939_DM3TX 169
J1939_ENABLE 139
J1939_GETDABYNAME 141
J1939_NAME 143
J1939_RX 150
J1939_RX_FIFO 151
J1939_RX_MULTI 153
J1939_SPEC_REQ 147
J1939_SPEC_REQ_MULTI 148
J1939_STATUS 145
J1939_TX 155
J1939_TX_ENH 156
J1939_TX_ENH_CYCLIC 158
J1939_TX_ENH_MULTI 160

© ifm electronic gmbh www.ifm.com

K

Klemme 15 ... 226
Konfiguration der Ein- und Ausgänge (Voreinstellung) ... 47
Konfigurationen ... 39

L

Laufzeitsystem ... 28, 227
Laufzeitsystem aktualisieren ... 41
Laufzeitsystem einrichten ... 39
Laufzeitsystem neu installieren ... 40
LED ... 227
LED im Anwendungsprogramm steuern ... 25
Leistungsgrenzen des Geräts ... 37
Link ... 227
LSB ... 227

M

MAC-ID ... 227
Master ... 227
MEM_ERROR ... 202
MEMCPY ... 203
MMI ... 227
Mögliche Betriebsarten Ein-/Ausgänge ... 216
MRAM ... 227
MSB ... 227

N

Netzwerkvariablen ... 58
NMT ... 228
Node ... 228
Node Guarding ... 228
Notizen • Notes • Notes ... 238

O

Obj / Objekt ... 228
Objektverzeichnis ... 228
OBV ... 228
OHC ... 205
OPC ... 228
operational ... 228
OUTPUT ... 183

P

PC-Karte ... 228
PCMCIA-Karte ... 229
PDM ... 229
PDO ... 229
PDU ... 229
PERIOD ... 178
PES ... 229
PGN ... 229
PID-Regler ... 229
Piktogramm ... 229
Piktogramme ... 7
Pre-Op ... 230
Prinzip der H-Brücke ... 190
Programmierhinweise für CODESYS-Projekte ... 30
Programmiersystem einrichten ... 42
Programmiersystem manuell einrichten ... 42
Programmiersystem über Templates einrichten ... 46
Prozessabbild ... 230
PWM ... 230
PWM1000 ... 185
PWM1000_LOW ... 187
PWM-Ausgänge ... 54

R

ratiometrisch ... 230
RAW-CAN ... 230
Reaktion abhängig von Betriebsart des Ausgangs ... 19
Reaktion auf Fehlermeldungen ... 213
Reaktion bei Einsatz von PWM oder CURRENT_CONTROL ... 19
Reaktion der Ausgänge auf Überlast oder Kurzschluss ... 19
remanent ... 230
Reset ... 34
Retain-Variablen ... 57
ro ... 230
RTC ... 231
Rücklesen von Retain-Variablen ... 57
RUN-Zustand ... 35
rw ... 231

S

SAE J1939 ... 138, 231
Schnelle Eingänge ... 51
Schnittstellen-Beschreibung ... 26
Schutzfunktionen der Ausgänge ... 18
SD-Card ... 231
SDO ... 231
Selbsttest ... 231
SET_IDENTITY ... 207
SET_LED ... 208
SET_PASSWORD ... 210
Sicherheitshinweise ... 10
Sicherheitshinweise zu Reed-Relais ... 24, 48
Sichern von Retain-Variablen ... 57
Slave ... 232
Software ... 27
Software-Filter der Ausgänge konfigurieren ... 52
Software-Filter der Eingänge konfigurieren ... 49
Software-Module für das Gerät ... 27
Software-Steuerungskonfiguration ... 43
Speicher, verfügbar ... 13
SRAM ... 13
Status-LED ... 25
Steuerungskonfiguration ... 43, 232
Steuerungskonfiguration aktivieren ... 44
stopped ... 232
STOP-Zustand ... 34
Stromregelung mit PWM (= PWMi) ... 55
Symbole ... 232

© ifm electronic gmbh
www.ifm.com

Systembeschreibung 12
Systemmerker 214
Systemvariable 232
Systemvariablen 46
Systemvoraussetzungen 12

T

Target 232
Target einrichten 43
TCP 232
Template 232
TIMER_READ_US 211

U

Über diese Anleitung 5
Übersicht
Dokumentations-Module für ecomatmobile-Geräte 6
UDP 233

V

Variablen 56
Verfügbarer Speicher 13
Verfügbarkeit von PWM 54
Verhalten des Watchdog 37
Verteilen des Anwendungsprogramms 33
Verwendung, bestimmungsgemäß 233
Vorkenntnisse 11

W

Was bedeuten die Symbole und Formatierungen? 7
Watchdog 37, 233
Welche Vorkenntnisse sind notwendig? 11
Widerstandsmessung 17
Wie ist diese Dokumentation aufgebaut? 8
wo 233

Z

Zykluszeit 233
Zykluszeit beachten! 31

© ifm electronic gmbh
www.ifm.com

10 Notizen • Notes • Notes

© ifm electronic gmbh www.ifm.com

© ifm electronic gmbh www.ifm.com

© ifm electronic gmbh www.ifm.com

© ifm electronic gmbh www.ifm.com

11 ifm weltweit • ifm worldwide • ifm à l'échelle internationale

Stand: 2015-03-06 8310

www.ifm.com • E-Mail: info@ifm.com

Service-Hotline: 0800 16 16 16 4 (nur Deutschland, Mo...Fr, 07.00...18.00 Uhr)

ifm Niederlassungen • Sales offices • Agences

D	ifm electronic gmbh Vertrieb Deutschland Niederlassung Nord • 31135 Hildesheim • Tel. 0 51 21 / 76 67-0 Niederlassung West • 45128 Essen • Tel. 02 01 / 3 64 75 -0 Niederlassung Mitte-West • 58511 Lüdenscheid • Tel. 0 23 51 / 43 01-0 Niederlassung Süd-West • 64646 Heppenheim • Tel. 0 62 52 / 79 05-0 Niederlassung Baden-Württemberg • 73230 Kirchheim • Tel. 0 70 21 / 80 86-0 Niederlassung Bayern • 82178 Puchheim • Tel. 0 89 / 8 00 91-0 Niederlassung Ost • 07639 Tautenhain • Tel. 0 36 601 / 771-0 ifm electronic gmbh • Friedrichstraße 1 • 45128 Essen
A	ifm electronic gmbh • 1120 Wien • Tel. +43 16 17 45 00
AUS	ifm efector pty ltd. • Mulgrave Vic 3170 • Tel. +61 3 00 365 088
B, L	ifm electronic N.V. • 1731 Zellik • Tel. +32 2 / 4 81 02 20
BR	ifm electronic Ltda. • 03337-000, Sao Paulo SP • Tel. +55 11 / 2672-1730
CH	ifm electronic ag • 4 624 Härkingen • Tel. +41 62 / 388 80 30
CN	ifm electronic (Shanghai) Co. Ltd. • 201203 Shanghai • Tel. +86 21 / 3813 4800
CND	ifm efector Canada inc. • Oakville, Ontario L6K 3V3 • Tel. +1 800-441-8246
CZ	ifm electronic spol. s.r.o. • 25243 Průhonice • Tel. +420 267 990 211
DK	ifm electronic a/s • 2605 BROENDBY • Tel. +45 70 20 11 08
E	ifm electronic s.a. • 08820 El Prat de Llobregat • Tel. +34 93 479 30 80
F	ifm electronic s.a. • 93192 Noisy-le-Grand Cedex • Tél. +33 0820 22 30 01
FIN	ifm electronic oy • 00440 Helsinki • Tel . +358 75 329 5000
GB, IRL	ifm electronic Ltd. • Hampton, Middlesex TW12 2HD • Tel. +44 208 / 213-0000
GR	ifm electronic Monoprosopi E.P.E. • 15125 Amaroussio • Tel. +30 210 / 6180090
H	ifm electronic kft. • 9028 Györ • Tel. +36 96 / 518-397
I	ifm electronic s.a. • 20041 Agrate-Brianza (MI) • Tel. +39 039 / 68.99.982
IL	Astragal Ltd. • Azur 58001 • Tel. +972 3 -559 1660
IND	ifm electronic India Branch Office • Kolhapur, 416234 • Tel. +91 231-267 27 70
J	efector co., ltd. • Chiba-shi, Chiba 261-7118 • Tel. +81 043-299-2070
MAL	ifm electronic Pte. Ltd • 47100 Puchong Selangor • Tel. +603 8063 9522
MEX	ifm efector S. de R. L. de C. V. • Monterrey, N. L. 64630 • Tel. +52 81 8040-3535
N	Sivilingeniør J. F. Knudtzen A/S • 1396 Billingstad • Tel. +47 66 / 98 33 50
NL	ifm electronic b.v. • 3843 GA Harderwijk • Tel. +31 341 / 438 438
P	ifm electronic s.a. • 4410-136 São Félix da Marinha • Tel. +351 223 / 71 71 08
PL	ifm electronic Sp. z o.o. • 40-106 Katowice • Tel. +48 32-608 74 54
RA, ROU	ifm electronic s.r.l. • 1107 Buenos Aires • Tel. +54 11 / 5353 3436
ROK	ifm electronic Ltd. • 140-884 Seoul • Tel. +82 2 / 790 5610
RP	Gram Industrial, Inc. • 1770 Mantilupa City • Tel. +63 2 / 850 22 18
RUS	ifm electronic • 105318 Moscow • Tel. +7 495 921-44-14
S	ifm electronic a b • 41250 Göteborg • Tel. +46 31 / 750 23 00
SGP	ifm electronic Pte. Ltd. • Singapore 609 916 • Tel. +65 6562 8661/2/3
SK	ifm electronic s.r.o. • 835 54 Bratislava • Tel. +421 2 / 44 87 23 29
THA	SCM Allianze Co., Ltd. • Bangkok 10 400 • Tel. +66 02 615 4888
TR	ifm electronic Ltd. Sti. • 34381 Sisli/Istanbul • Tel. +90 212 / 210 50 80
UA	TOV ifm electronic • 02660 Kiev • Tel. +380 44 501 8543
USA	ifm efector inc. • Exton, PA 19341 • Tel. +1 610 / 5 24-2000
ZA	ifm electronic (Pty) Ltd. • 0157 Pretoria • Tel. +27 12 345 44 49

Technische Änderungen behalten wir uns ohne vorherige Ankündigung vor.
We reserve the right to make technical alterations without prior notice.
Nous nous réservons le droit de modifier les données techniques sans préavis.

© ifm electronic gmbh www.ifm.com